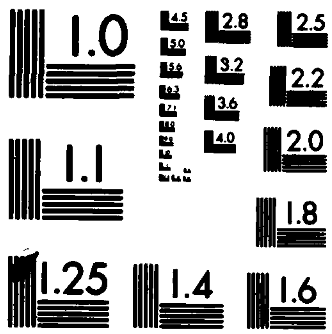
AD-A138 226   MULTIBUS AVIONIC ARCHITECTURE DESIGN STUDY (MAADS)(U)   1/2
TRW DEFENSE SYSTEMS GROUP RDDONDO BEACH CA
B A RICH ET AL. OCT 83 AFWAL-TR-83-1141

UNCLASSIFIED   F33615-81-C-1520   F/G 17/2   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

AFWAL-TR-83-1141

AF WRIGHT AERONAUTICAL LABORATORIES

MULTIBUS AVIONIC ARCHITECTURE DESIGN STUDY (MAADS)

TRW Defense and Space Systems Group
One Space Park
Redondo Beach, California 90278

October 1983

Final Report for Period 21 September 1981 to 30 September 1983

Approved for public release; distribution unlimited.

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Claude M. Fletcher, Jr

CLAUDE M. FLETCHER, Jr.
Technology Applications Group
Avionics Laboratory

Kenneth N. Frankovich

KENNETH N. FRANKOVICH, Capt, USAF
Chief, Technology Applications Group
Avionics Laboratory

FOR THE COMMANDER

Raymond D Bellem

RAYMOND D. BELLEM, LT COL, USAF
Deputy Chief
System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/AAAS-1 *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWAL-TR-83- 1141	2. GOVT ACCESSION NO.: AD-A138 226
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Multibus Avionic Architecture Design Study (MAADS)	5. TYPE OF REPORT & PERIOD COVERED: Final Report 21 Sept. 81 - 30 Sept. 83
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): B. A. Rich, J. L. Stautberg, D. G. Haldeman, W. P. Whalen	8. CONTRACT OR GRANT NUMBER(s): F33615-81-C-1520
9. PERFORMING ORGANIZATION NAME AND ADDRESS: TRW Defense Systems Group, One Space Park, Redondo Beach, California 90278	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: P.E. 62204F 2003-09-22
11. CONTROLLING OFFICE NAME AND ADDRESS: Avionics Laboratory (AFWAL/AAAS), Air Force Wright Aeronautical Laboratory, Wright-Patterson AFB, Ohio 45433	12. REPORT DATE: October 1983
	13. NUMBER OF PAGES: 139
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Avionics Software	DAIS	MIL-1750
Avionics Systems	High Speed Bus	Multiplex
Avionics Architecture	MIL-STD-1553	Standardization
ASID	MIL-STD-1589	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Multibus Avionic Architecture Design Study (MAADS) evaluated projected avionic requirements for tactical aircraft of the 1990s and defined an architectural approach and design example suitable for use as the baseline for the Avionic System Integration Demonstrator (ASID) System Definition project. The architectural approach is multi-bus in nature including MIL-STD-1553B bus, a high speed bus, and a video bus. System sizing and timing estimates are provided. Areas for potential future standardization are identified.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

During the past decade the Air Force has been developing and studying architectural approaches for avionic systems for advanced tactical aircraft. These efforts, typified by the Digital Avionics Information System (DAIS) program have expedited the development of three Military Standards for avionics architecture design. These are MIL-STD-1553B (Multiplex Data Bus), MIL-STD-1750A (Computer Instruction Set), and MIL-STD-1589B (JOVIAL Software Language). The objective in developing these standards was to provide cost and logistic advantages inherent in standardization and to support the development of high performance, easily expandable avionic system architectures.

Even as these standards were maturing and being successfully implemented in a variety of fielded systems, the Air Force recognized that new technologies are becoming available and conditions indicate that avionic mission requirements for the 1990s timeframe will be more stringent than those currently encountered. The Air Force therefore initiated this Multibus Avionic Architecture Design Study (MAADS) to design next-generation tactical avionic architectures with a view to exploiting the benefits of standardization achieved to date, accommodating the infusion of new technologies as they become available, and identifying potential new areas of standardization. The avionic architectures developed by the MAADS program are intended to be considered for implementation by the Advanced System Avionics project at the Avionics Laboratory. The MAADS program may thus be viewed as a key lead in to the PAVE PILLAR program being undertaken by the Air Force Wright Aeronautical Laboratory (AFWAL).

The study was performed in a multi-phase fashion. In the first phase, government provided mission specifications for projected 1990s tactical missions were reviewed and evaluated for their impact on avionic architectures. The study was directed to take a "quick look" at the implications the long term mission requirements had on the relatively near term technology thrusts being pursued by the Air Force. This phase concluded in a System Requirements Review (SRR) and presented an approach to architecture specifications, some potential architectures, preliminary system sizing information, and a list of future partitioning issues. It resulted in the identification to the Air Force of avionic processors equipped with multiple channels/smart channels as the key technology area requiring immediate attention. It identified the laboratory simulation support facility as an area of concern.

The second phase of the study expanded the scope and the depth of the analysis and culminated in the System Assessment Review (SAR). This review updated the results of the previous review. It also identified and sized the variety of data bases required to support future avionics missions. It presented a "strawman" architecture for future systems and scoped the issues associated with the definition of a high speed data bus.

The third phase of the study continued to enhance the level of detail addressed. It culminates with this final report but was also the subject of an Interim Technical Review (ITR). It produced a Mission/Environment Specification, a System Requirement Specification, an Architecture Specification, a Computer Program Development Specification (CPDS) for avionics real time operating system software, and an informal report documenting the analysis of high speed bus protocols. The architecture specification covered the Operating System to Application Software interface control definition (ICD), system control procedures, a high speed bus definition, a 1750A processor high speed bus control interface definition (ICD), a 1750A processor 1553B bus control interface definition (ICD), a mass memory subsystem interface definition and a standard backplane interconnect definition.

In the course of the study the effort was redirected and augmented to some extent. The early guideline that only evolutionary enhancements to existing standards and technology be considered was removed. The study was specifically directed to not be constrained by existing standards and technology. Nonetheless, the study conclusions show that a large portion of needed improvements can be accomplished via evolutionary enhancements. The task of implementating additional required functions was particularized to consist of upgrading the AFWAL avionics executive software to 1750A and rehosting it on a VAX 11/780 computer.

2.0 BACKGROUND

The Avionics Laboratory of the Air Force Wright Aeronautical Laboratories (AFWAL) has a continuing mission to explore the future requirements of avionic systems and to encourage, accelerate and channel technology developments so as to assure those requirements can be met. In the context of this continuing mission, the MAADS project may be viewed as the bridge from the DAIS program of the 1970s to the PAVE PILLAR program of the 1980s.

The DAIS program was undertaken in response to a situation in which each avionics system was essentially a point design incorporating technology available at the time and implemented primarily in analog hardware. Such systems were difficult to maintain and upgrade. DAIS demonstrated that implementation in digital processors was feasible and that with a proper division between executive software and application software a high degree of flexibility could be achieved. The architecture of multiple avionic processors interconnected and communicating via a MIL-STD-1553 multiplex bus, and controlled by executive software written in JOVIAL, a high order language (HOL), has gained widespread acceptance. Whether derived from DAIS, or independently developed, these architectural concepts will be found underlying most modern day avionics system developments. The DAIS effort also expedited the development of military standards in the areas of the multiplex bus (1553B), the computer instruction set architecture (1750A), and the high order language (1589B). The future cost savings of interoperable and transportable software, more competitive procurements, and more easily maintained equipments resulting from these standards are almost inestimable. The success of the DAIS program in furthering vital Air Force objectives is demonstrable and significant.

But, the Air Force cannot rest on these accomplishments. The PAVE PILLAR program is directed at defining the avionic systems for tactical aircraft in the 1990s. Examination of future requirements indicate that the number of aircraft available for missions will be fewer. Furthermore, emphasis is now being placed on operating aircraft at remote locations where maintenance support equipment will not be available. In addition, air defense systems expected to be encountered will be considerably more sophisticated and more dense. Lastly, tactical aircraft operations at night, under the weather, are called for.

The PAVE PILLAR objective is therefore to sharply improve performance for more complex missions facing more difficult obstacles and do so at a reduced cost. Key to the achievement of this objective is increased survivability, fault tolerance and reliability as well as exploiting opportunities to provide optimal mission performance by coupling formerly uncoupled systems and fusing information from currently independent functions.

Pursuant to these objectives the Air Force has sponsored a number of projects to investigate specific aspects of the required technologies:

- The Fault Tolerant Computer Network Study (performed by IBM under contract F33615-79-C-1108) evaluated fault tolerant techniques in general, and specifically, assessed the fault tolerance characteristics of the DAIS.

- The Integrated Testing and Maintenance Technology study (performed by Boeing under contract F33615-81-C-1517) defined requirements and specifications for implementing integrated testing and maintenance concepts.

- The Advanced Aircraft Electrical System Control Technology Demonstration project (performed by Boeing under contract F33615-80-C-2004) provided information on integrating the aircraft electrical system control.

- The Advanced Avionic Systems for Multimission Applications study (performed by Boeing under contract F33615-77-C-1252) examined multibus architectures and alternate bus control schemes. 82-1252,82-1076,82-1076.

- The Multifunction, Multiband Airborne Radio System (performed by TRW under contract F33615-77-C-1172) investigated integrating communications into single subsystems. (AFWAL-TR-81-1113)

- The Integrated Communication, Navigation, and Identification Avionics (ICNIA) program (multiple contractors) is pursuing an even more highly integrated subsystem.

- The Integrated Electronic Warfare System (INEWS) (multiple contractors) is defining a similar highly integrated approach in the EW area.

- The Very High Speed Integrated Circuits (VHSIC) program (multiple contractors) is seeking an order of magnitude or more increase in processing speeds.

In addition to the Air Force sponsored research there exists widespread interest and research in the area of computer networks and high speed bus technology, especially as related to fiber optics. The objective of the MAADS effort becomes one of assimilating and consolidating the results of the above efforts along with other current research efforts in avionics in order to define the architectural approach for future avionic systems.

3.0 MAADS PROGRAM REVIEWS

The MAADS effort was organized into a series of analysis phases, each progressively more broad in scope and more detailed in nature and culminating in a major project review with the Air force. The final wrap-up period of the study also incorporated the upgrading and rehosting of the AVSAIL executive software. The following sections report the study activity in more detail.

3.1 System Requirements Review

3.1.1 General Requirements

As noted earlier, this phase of the study was directed at taking a quick look analysis of future mission requirements with a view to identifying issues which could potentially influence near term procurements. Of specific concern were potential impacts on the Advanced System Avionics (ASA) processor design, the Advanced Digital Avionics Map (ADAM) functionality and I/O requirements, and facility support requirements.

The effort began by reviewing the broad scope of future mission requirements and noting the rapidly evolving and expanded use of advanced technology, the ever increasing threat density and the increased demand for mission responsiveness. The objectives include the increased availability, reliability, and survivability of the avionics system and the aircraft as a whole. The tactical aircraft must be a multi-mission system with the worst case requirements defined by the all-weather night mission. A dramatic increase in system functionality is to be expected and yet space, weight and power constraints must be met. The system must be easily adaptable to new threats, new missions, and the injection of new technologies. These include both advances in digital hardware technology and integration aproaches as represented by the trends to integrated maintenance and data fusion algorithms. The system must continue the thrust toward increased automation to offload the pilot and, in general, must accommodate increased system sophistication while simplifying the pilot vehicle interface.

3.1.2 Sizing And Processing Estimates

With this broad viewpoint as background, a list of major system functions was generated and sizing and processing estimates were made, as shown in Table 1.

Table 1. MAADS SRR Data Processing Requirements Estimates

FUNCTION	MEMORY (kBYTES)	THROUGHPUT (MIPS)	REMARKS
Executive	40/Processor	.05/Processor	Modified DAIS Executive, Distributed Bus Control
Electrical Power Control	36	.075	AAES
Fault/Detection Redundancy Management	30	.1	Estimate
Integrated Maintenance	25	.05	Estimate - OFP Only
EW	96	.16	TRW NTWS Study
Control/Display	300	.5	~DAIS (MPDG & Appl.) x 2
Stores Management	30	.01	Estimate
ICNIA	227 (434)	1.2 (3.26)	TRW (ITT) Design Estimate
Integrated Navigation	22 (50)	.09 (.13)	TRW (ITT) Design Estimate
Tercom	20	.05	Estimate
Mission Control	100	.1	Estimate
Profile Planning			
4D/Energy	32	.25	Consistent with LSI/KC-135
Path Optimization	1500*	2-5**	Includes Threat Masking
TF/TA Coupling	8	.02	
IFFC	144	.4	Systems Control, Inc.
Target Acquisition			
FLIR Sensor/Tracker	64	.25-.5	
Image Processing	12000*	40**	Per Sensor
Fusion	---	4**	Per Sensor
Digital Map Display	1900*	1.5-3**	AETMS
IFF Fusion	?	10-100**	Paper - "Computational Consideration for fusion in target identification system" E. L. Waltz, Bendix

*Essentially Data Storage. **Processing lends itself to Signal Processing

Functions assigned to subsystems in existing system designs are not included in Table 1, whether signal processing or data processing functions. Some of the functions listed are obvious candidates for special purpose processing due to their high throughput requirement and the nature of the processing required. The intent of this exercise was to identify representative memory and throughput requirements to aid in further partitioning studies. During the SRR, it was noted that the terrain display processing may require considerably more throughput than the 1.5-3 MIPS listed.

3.1.3 Architectural Considerations

Along with the "ball park" estimates of system sizing and processing requirements, a general review of architectural considerations was undertaken. First, a "shopping list" of architectural issues was drafted as follows:

- Building Blocks
- Connectivity
- Control Strategy
- Processing and Mass Memory Partitioning
- Throughput and Memory
- Technology Injection
- Flexible/Extendable
- Common/Standard Interfaces
- Standardization Evolution

It was recognized that the quick look analysis could not explore all these issues or pursue questions at great length. Nonetheless, it was felt by highlighting these items early, major impacts would be identified and a workable architectural approach could be defined which could be adjusted and refined as more detailed analysis was performed at a later time.

In general, a system architecture is fully specified by defining the system elements (components) to be utilized, how they are interconnected (topology), and the control strategy employed to manage the overall systems operation. Each of these areas will be discussed more fully later. In addition, the architecture establishes the upper limits on the computational and data storage capacity of the system and, hence, must be sized with adequate processing, memory and data flow to meet system/mission requirements. The architecture must also address system level issues such as backup, reconfiguration, and degraded mode operations. The architecture does not resolve these issues per se, but must, rather, provide the context in which one or more alternate solutions can be implemented.

The core system elements expected in a future avionics architecture are: processors, both general purpose and signal/parallel processors; data buses, both 1553B and yet to be defined higher speed buses; mass memories; controls and displays; data base transfer units; and Executive system software.

Following the identification of the components the next architectural consideration was the interconnection of the data processors via the data buses. That is not to say that the subsystems are unimportant, but the subsystem interconnections are driven by specific data flow and fault tolerance considerations appropriate to the subsystem.

An area requiring special future attention is the connections between data processors and either mass memory units and/or signal/parallel processors. These connections are likely to be required to handle large volumes of data at very high data rates which may be significantly beyond the capability of 1553B.

A brief survey of topologies such as rings, stars, shared memory, etc. revealed the multiplex bus approach strikes a good compromise among the desirable system characteristics of modularity, expandability, fault tolerance, and good throughput. Alternate topologies tend to be characterized by excessive hardware, complex control strategies and/or poor fault tolerance.

The Air Force has committed itself to multiplex techniques and the study recommends continuing in that vein, expanding throughput with additional buses. In addition, large numbers of subsystems now built or currently being built are compatible with 1553B and are going to be required to be integrated into the system.

The multitiered bus expands the modularity, expandability, and the data throughput capabilities of the global bus. There are two basic types of multitiered buses: the hierarchical and the parallel.

The hierarchical configuration may be visualized as tree-like (see Figure 1) with a global bus and a number of sub-buses which operate more or less independently. It is generally characterized by good throughput (the sub-buses operate simultaneously), good fault tolerance (a failure of one sub-bus does not affect another), and excellent growth potential (sub-buses may be added almost indefintely), but exhibits poor reconfiguration capabilities. The parallel topology is one in which all processors are connected to all buses. It also has good throughput and fault tolerance characteristics and provides a dramatically increased flexibility as compared to the hierarchial approach.

The study observed that it is very important in a hierarchial approach that the functional partitioning match well with the physical partitioning. Otherwise the independance of the sub-buses is compromised and the consequent benefits lost. Also if communication of a unit on one sub-bus is required with another unit on a different sub-bus some very complex message sequences are needed. The implications to control strategies and error management are severe.

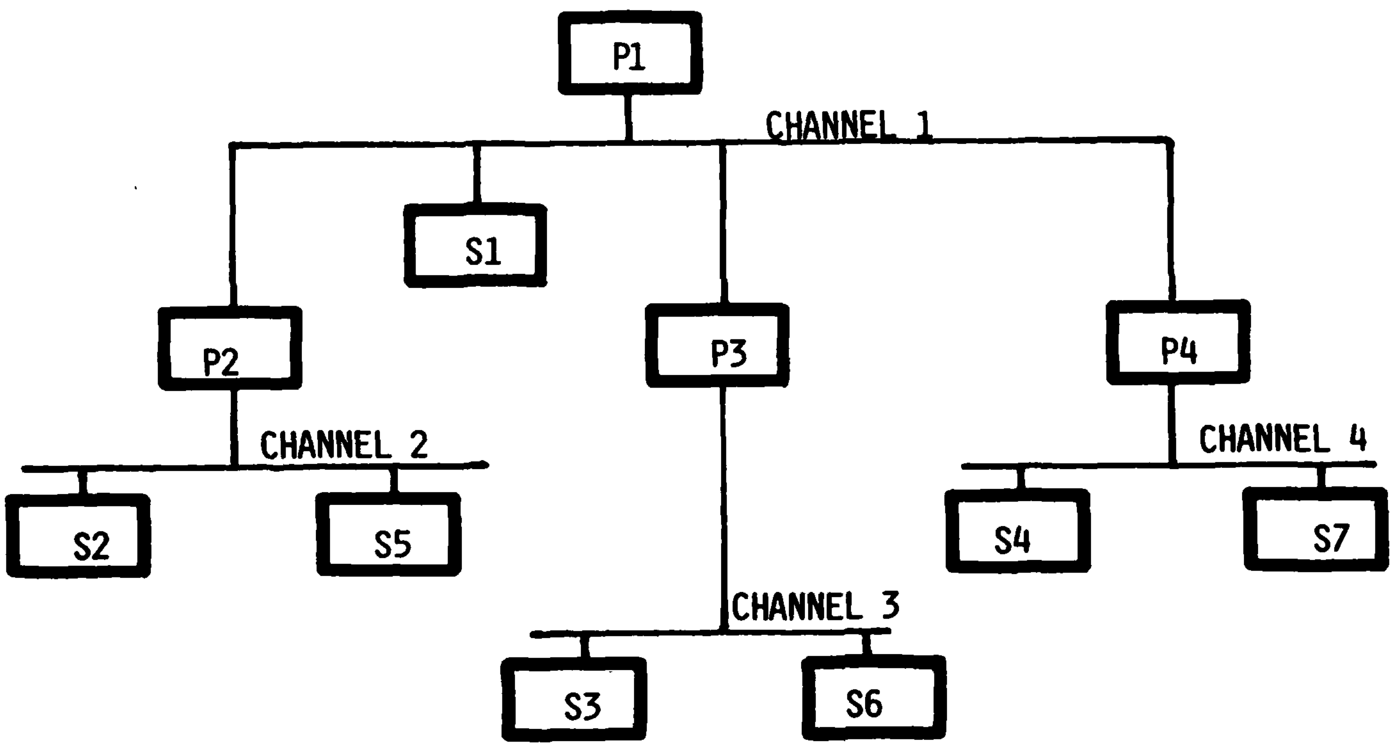


Figure 1. Hierarchical Configuration

A second key observation was that the fault tolerance shortcomings of the hierarchical topology are typically addressed by adding interconnections (i.e., cross strapping) sufficient to permit a processor to assume some or all of the functions of another processor in the event of failure. As this methodology of cross strapping processors on sub-buses is continued, the hierarchical configuration begins to take on the appearance of a parallel, multi-bus configuration. In the limit of achieving full functional reassignability of all processors the hierarchical configuration becomes the parallel topology.

Based on these observations, the study recommended that the full connectivity be provided to support the parallel bus structure. Then, by selectively activating the channels in the various processors a specific system can implement a simple hierarchical structure, a fully symetric parallel structure or whatever in-between configuration is desired for the specific application. Figure 2 illustrates this point with the solid lines indicating an active channel. The configuration is actually identical to that in Figure 1.

A final observation was that this parallel topology is so as seen from the top level, i.e., system processor, viewpoint which, in effect, treats subsystems as "black boxes". When more complex subsystems are examined in more detail, what is found is an internal hierarchical structure, as depicted in Figure 3. The conclusion drawn is that the overall system architecture of future avionics systems will be a hybrid, appearing parallel at the system or mission level and hierarchical at the subsystem level where functional isolation permits.

With respect to the third element of system architecture, control strategy, the study at this point identified a number of candidates for further analysis. It was noted that the flexible parallel topology would permit MAADS to investigate a wide variety of control strategies. It was indicated that special attention would be paid to distributed control strategies since it is to be expected they will better support fault tolerance objectives.

Another benefit of the selection of the parallel topology is that it readily supports the insertion of new technology. Several examples were reviewed and it was shown that the architecture provides a framework within which the new technology impacts may be clearly and specifically represented.

The important implication of the selected parallel topology is that future avionics LRUs, especially avionics processors, will require multiple channel connections. The study further concluded that to avoid bogging down the processors with channel management functions it is imperative that very intelligent channel interfaces be the norm. The near term procurement of processors should, therefore, include the specification for a "smart BCI".

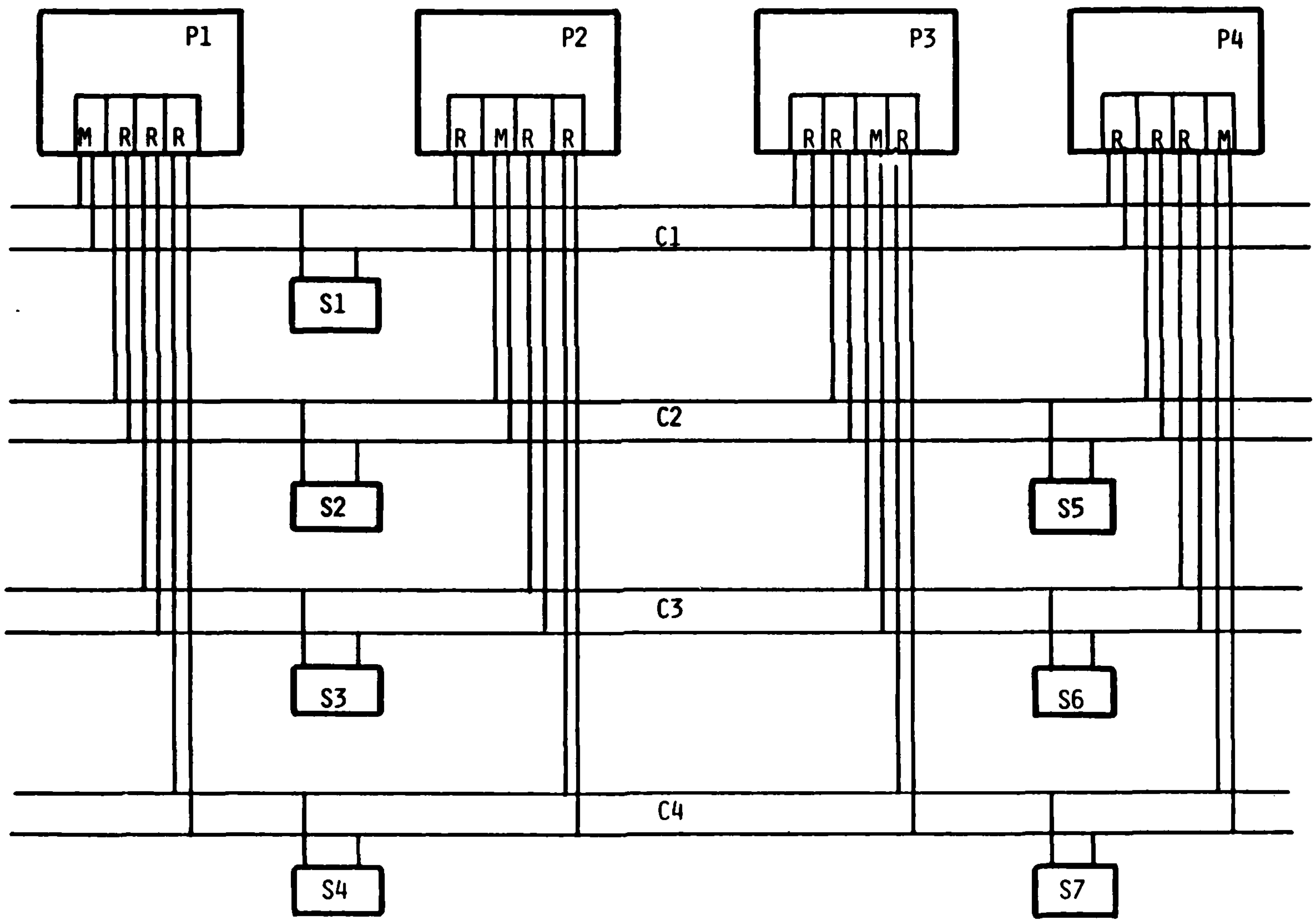


Figure 2. Selective Activation of Channels

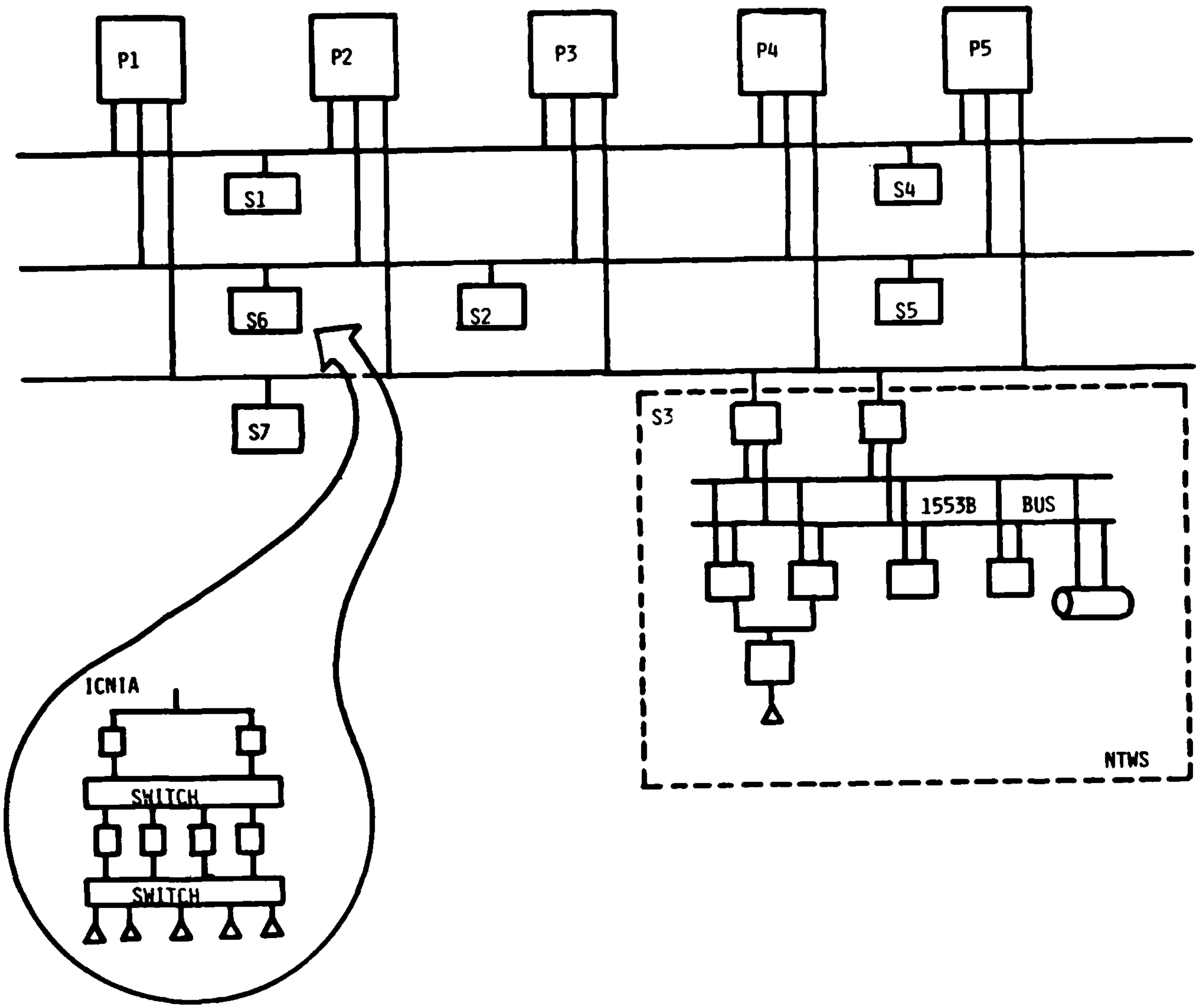


Figure 5. Parallel/Hierarchical Configuration

3.1.4 Bus Control Interface (BCI) Specification

A particular BCI specification was then reviewed. This specification was an output of the Programmable Interface for Multiplex Systems (PIMS) program and was currently under consideration for use with Embedded Avionics Processor (EAP). The highlights of this BCI specification are that it is supported by a high level command structure with functions such as Initiate, Terminate, Resume, and Clear, and contains sufficient working register and microprogramming to operate quite autonomously from the processor once initiated. This includes an exception chaining capability that allows the BCI to extract a "program" from processor memory and deal with errors and exception conditions that otherwise would require a processor software interrupt handler.

3.1.5 Data Bases / Mass Memory Issues

The next study topic reviewed was that of system level data bases and related mass memory issues. An early study conclusion was that there would be many different functional data bases aboard the future tactical aircraft. These include integrated maintenance data, mission data, threat data, terrain data and software loads to support reconfiguration. The study identified issues to be addressed including the partitioning of data bases and allocation of mass memories.

Other issues with the Mass Memory include type of memory (tape, disk, bubbles) and how it is tied into the architecture (dedicated subsystem bus, general high speed system buses, 1553B). Potentially, more than one subsystem may need access to the data. This will affect the control/access strategy that must be developed as well as the data formatting problems and data security problems.

Closely related to these issues of the data base/mass memory is the problem of reprogramming the data base to account for changes in the threat data base, new mission flight plans and new targets. This reprogramming capability must be able to be accomplished in-flight as well as pre-flight. Its wide-range affects include data base generation, data formatting, data entry techniques, and data base update management. Pilot involvement in the reprogramming of the data bases during flight must also be considered.

3.1.6 Interim Demonstrations

At this point in the System Requirements Review a slight digression was taken to point out that with only moderate effort some system characteristics could be demonstrated and issues investigated in the laboratory well before the timeframe of the planned demonstrations. Key topics to be considered include:

o dual channel architecture

- executive software improvements
- restart of application software
- data base management approach
- hot and cold backup
- terminal failure and recovery

3.1.7 Partitioning Considerations

The next section of the review addressed the problems and issues associated with the partitioning of data processing tasks between the system or central core avionics and the subsystem. Following a brief overview on the diverse types of subsystems considered and the overall trends in subsystem development, several different approaches to the overall partitioning problem were investigated (by way of examples) against a set of partitioning criteria. On the basis of these examples, some general observations were made regarding the various approaches to the system/subsystem partitioning problem.

In general, MAADS observed a trend toward subsystems which are more complex and, correspondingly, more software intensive. To a large extent, this trend is driven by economic factors such as: 1) the reduction of size, weight and life cycle costs via the efficient use of hardware and signal processing and 2) the ability to support new technology and adapt to new requirements in a cost effective manner.

Key issues which must be considered by each subsystem are the partitioning problems internal to the integrated subsystem, the partitioning problems between the subsystem and the central avionics system, and the physical partitioning of the subsystem. For example, subsystems such as ICNIA and New Threat Warning System (NTWS) may be viewed as systems in and of themselves from both a functional and a physical point of view. Conversely, functions such as fault tolerance, integrated maintenance and EMUX may be overlayed on the avionics suite and, as such, can only be viewed as separate entities from a functional point of view.

Table 2 depicts the various levels of processing that are required in many of the advanced avionic subsystems and illustrates that the internal partitioning of the processing tasks within a subsystem is primarily driven by technology constraints. At one extreme, the preprocessor or special purpose digital signal processing circuitry provides a very high signal throughput and a minimal degree of programmability. At the other extreme, the data processor provides a relatively low signal throughput and a high degree of programmability. As advancements are made in digital technology, the trend will be for each of these three generic processors to perform tasks that require higher signal throughput with an increasing degree of programmability and memory. That is, the preprocessor will become increasingly involved in those signal processing tasks that are now relegated to the front-end hardware and, similarly, the data processor will become increasingly involved in both the data processing functions and those tasks that are now performed by the programmable signal processor.

In discussing the problems associated with the partitioning of processing tasks between system and subsystem, it should be kept in mind that a subsystem may be one of several physical configurations. Furthermore, when considering any one subsystem, the global or system avionics may include one or more other subsystems of the avionics suite (and may, therefore,already include requisit processing).

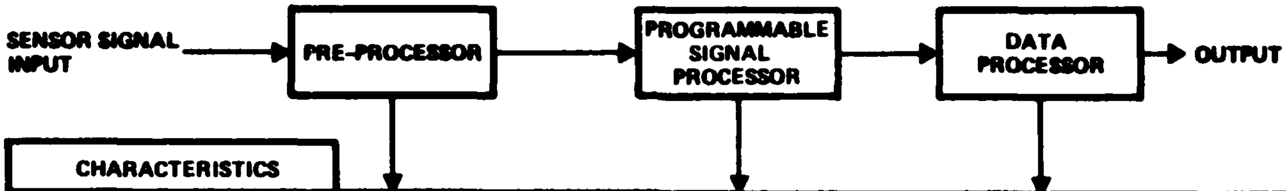
There are three primary types of partitioning problems. The first type involves those processing tasks associated with the control of a subsystem and the second type involves those processing tasks associated with the fusion of data either within an integrated subsystem or among multiple subsystems. Finally, a third type of partitioning problem involves the consideration of fault tolerance and the physical configuration of the subsystem.

The four basic criteria that were used to evaluate the various approaches to system/subsystem partitioning of data processing tasks are bus loading, standard interfaces, subsystem autonomy, and fault tolerance.

Bus loading is an obvious criterion since the connectivity between system and subsystem must be capable of supporting the overall throughput requirements. Although the application of a wideband bus/high speed memory would tend to de-emphasize this criterion, the constraints of MIL-STD-1553B will continue to be present for several years since many of the existing and future subsystems will be designed to this interface.

It is desirable from an economic standpoint to partition processing tasks between the system and subsystem to facilitate the definition of a standard interface (witness development of the INS). If a stable,standard interface can be defined, the interchangeability of a subsystem in an aircraft or, conversely, the use of a subsystem in several different aircraft can be accomplished with minimal impact. Furthermore, as technology progresses, the subsystem can be modified/enhanced without affecting the processing within an avionics system.

Table 2. Subsystem Partitioning



CHARACTERISTICS			
• SIGNAL THROUGHPUT RATE	VERY HIGH 100 TO 10,000 MOPS	HIGH 10 TO 1,000 MOPS	LOWER 0.1 TO 10 MOPS
• TYPE OF PROCESSOR ARCHITECTURE	SPECIAL FUNCTION UNIT (SFU)	MULTIPROCESSOR, ARRAY PROCESSOR, PIPELINE PROCESSOR, PARALLEL PROCESSOR	UNIBUSS GENERAL PURPOSE
• DEGREE OF PROGRAMMABILITY	LOW (99 PERCENT HARDWARE)	MEDIUM EXTENSIVE (50 PERCENT HARDWARE)	EXTENSIVE (10 PERCENT HARDWARE)
• TYPES OF DIGITAL LSI	VERY HIGH SPEED, LOGIC INTENSIVE	HIGH SPEED, LOGIC AND MEMORY	MEMORY INTENSIVE
• OTHER CHARACTERISTICS	• APPLICATION-DRIVEN, STANDALONE CUSTOM CHIPS • REGULAR FLOWFORM ALGORITHMS	⟶	• STANDARDIZED AND SEMI-CUSTOM CHIPS • RANDOM ALGORITHMS, BRANCHING, STORED PROGRAM

Another criterion for partitioning between system and subsystem is in the support of the self test and fault isolation functions of integrated maintenance. If the outputs of a subsystem are meaningless without additional data processing, testing and operation of the subsystem must be performed in concert with the system avionics. Therefore, it is desirable for some degree of system autonomy to support standalone operation and test.

Fault Tolerance is a partitioning criterion that delves into both the functional and physical aspects of an avionics system. From a functional point of view, the failure of a subsystem or part of a subsystem should not restrict the ability of the system to support the avionics function in a degraded mode capacity. The results of examining the partitioning issues for several specific subsystems proved to be fairly common and may generally be described as follows;

(a) When all processing is moved in the core avionics, maximum bus loading results and a poor-to-fair potential for a standard interface definition exists. This approach tends to inhibit technology improvements related to subsystem control and generally inhibits subsystem interchangeability. A poor subsystem autonomy results, making a unit difficult to test without other avionics functions. This approach does yield good fault tolerance since the core software can react to partial subsystem failures and institute degraded mode operations.

(b) When all processing is moved into the subsystem, minimum bus loading results. The ability to standardize the interface is still rated only poor-to-fair and this inhibits aircraft interchangeability. Fair-to-good system autonomy results and standalone testing is feasible. Fault tolerance now is rated very poor since a subsystem failure tends to result in a total loss of function.

(c) When the processing is split between the subsystem and core avionics and the split is based on the distinction between mission oriented and subsystem oriented functions very good results are achieved. Bus loading is only moderate. A standardized interface is feasible allowing subsystem/aircraft interchangeability while still supporting technology growth. Fair-to-good subsystem autonomy is achieved and good fault tolerance results.

As a practical matter, partitioning must be evaluated for each subsystem on an individual basis and will be constrained by the signal processing/pre-processing technology. Within those constraints, the partitioning should be on a functional basis so as to enhance the potential for a standard interface.

By way of example,this partitioning approach was applied to the terrain data processing functions and it was shown that a threefold division is indicated: Display Processing, Terrain Data Base Subsystem, and Penetration Profile Subsystem.

The earlier memory sizing and throughput estimates were then revisited and selected functions removed due to their assignment to subsystems. Table 3 shows the resulting requirements on central data processing.

3.1.8 Demonstration Strategy

The final topic of the System Requirements Review was a brief discussion of demonstration strategy and simulation requirements. The following conclusions were offered:

- In addition to real-time hot bench demonstration, analysis and non-real-time simulation should be used to validate advanced avionics technology. In general, the latter two methods are more effective in validating performance. Real-time demonstrations of selected functions and/or mission segments are especially effective where significant pilot interaction is required.

- The nature of the avionic capabilities to be validated requires facility enhancements. A real-time monitor is proposed to provide detailed system status displays and operation control. The scenario generator provides an option of pilot-less operation to provide repeatability and ease of operation. A standard software test environment is proposed to allow various users access to common interfaces and software tools. In anticipation of extensive facility use by real-time and non-real-time users, it is recommended that processor resources be increased.

- Required simulation enhancements include new sensor models, new outputs from existing models (e.g., BIT outputs) and both error and failure mechanisms which will modify existing idealized sensor output messages.

3.2 System Assessment Review

3.2.1 General

The second phase of the MAADS effort expanded the scope and depth of the analysis as planned but was also driven to some extent by the Air Force customer indicating an urgent need for a "strawman" architecture and the recognition that interest and activity in the area of high speed bus technologies was escalating rapidly. The System Assessment Review thus contained a distinct emphasis on estimating data communication requirements and data base requirements.

Table 3. Central Data Processing Estimates

FUNCTIONS	MEMORY (KBYTES)	THROUGHPUT (MIPS)
EXECUTIVE	40N	.05N (N PROCESSORS)
POWER CONTROL	36	.075
FAULT DET./RECON.	30	.1
INT. MAINT.	25	.05
TERCOM	20	.05
MISSION CONTROL	100	.1
PROFILES (4D/ENERGY)	32	.25
TF/FA COUPLING	8	.02
IFFC	144	.4
SENSOR/TRACKER	64	.4
TOTAL	459 + 40N	1.45 + .05N

MEMORY	258K WORDS	THREE AYK-15A (~.5MIPS)	PROCESSORS
	238K WORDS	OR TWO .8 MIPS PROCESSORS	
	218K WORDS	OR ONE 2 MIPS PROCESSORS	

NOTE: DOES NOT INCLUDE BACKUP PROCESSORS

The architecture and strawman configuration were constrained by existing and anticipated technology, as well as by existing and pending subsystem configurations. Schedule constraints were such that a "snapshot" had to be taken on which to base a strawman system. The architecture had to accommodate 1553B interfaces and AN/AYK-15A processors and yet be flexible enough to accommodate yet to be defined high speed buses and advanced processors. Subsystem designs existed for ICNIA, NTWS, ADAM (but in a redifinition phase), stores and controls and displays.

Since quantitative mission requirements had not been provided, assumptions had to be made. Where the level of performance was related to system configuration, the effect of alternative assumptions was considered. In general, worst case assumptions were made so that the strawman configuration would not limit system performance.

3.2.2 Influence Of Other Projects

Results of other studies were considered. From a preliminary examination of available results, no significant architecture impacts were expected. Appropriate processing throughput and memory were allocated as required for power control and integrated maintenance. In general, the strawman configuration was consistent with the Fault Tolerance Study conclusions,and could be implemented with the PIMS designed BCI.

The Advanced Aircraft Electrical System Technology Demonstrator (AAESTD) study, performed by Boeing concluded the function could overlay the avionics with no architectural impacts. The Fault Tolerant Computer Network Study (FTCNS), performed by IBM identified some localized enhancements and concluded a single spare system processor was still the best approach for enhanced fault tolerance. No solution to the monitor contents problems was identified and the cost effectiveness remained a total system rather than a specifically avionics question. The Integrated Test and Maintenance Technologies Study (ITMTS), being performed by Boeing indicated a distributed function with no architectural impacts. The Programmable Interface for Multiplex Systems (PIMS) effort, performed by Sperry-Univac/TRW, had provided an approach to the smart channel requirements.

3.2.3 Function Definition And Partioning

The methodology used to derive architectural factors from mission requirements is to take the mission requirements, perform a functional synthesis, define major subsystems and then evaluate the implied data communication requirements, data processing requirements and data bases. Functional synthesis is an iterative process. Partitioning is indicated by processing and data flow requirements. Conversely, a partitioning must be assumed in order to assess data communication requirements. The mission/system requirements relationship is depicted in Table 4. The mission/system requirements noted in the left hand column are specifically mentioned in the MAADS SOW. The related avionic functions in the right hand column are implied.

The profile optimization process should include considerations of survival (via threat avoidance, terrain avoidance, terrain masking), mission effectiveness (some target value factored by success probability and added to survival probability), ECM allocation, weapon allocation, time of arrival, fuel, threat location uncertainty, and aircraft speed. The profile optimization processing task is probably well beyond (tens of MIPS) the throughput capabilities of available general purpose data processing technology. In-flight terrain masking of newly discovered threats may not be feasible due to threat location uncertainties and response time requirements.

An image processing function is envisioned (perhaps an image processing subsystem) as represented in Figure 4. The terrain display function has been included since terrain image is potentially important to image fusion. A display is constructed for pilot viewing from the best composite image and the results of image analysis.

Several complex estimation functions must be performed such as target location data fusion, threat location data fusion and IFF fusion. Accurate target location allows NAV bombing, improves survivability, standoff weapon effectiveness, and overall mission effectiveness. The need for a fast response may dictate special purpose processing hardware which is capable of a very high throughout for a moderaste amount of data. Accurate threat location is needed for threat masking. IFF fusion must provide spatial and temporal track correlation and deal with multisource data including direct active (ICNIA), direct passive and indirect (JTIDS). Rapid classification with a high degree of confidence is needed.

An integrated guidance/weapon delivery function is envisioned as depicted in Figure 5. Given the detailed profile planning process, guidance is straightforward. An "inner loop" is provided by active target tracking which augments the nominal mission plan. The Continuously Computed Impact Point (CCIP) computation provides a basis for time-to-release for weapons and outputs a time-to-go which allows the stores management subsystem to compensate for any internal delays.

A high level system functional diagram emphasizing terrain data management is shown in Figure 6. The functions in heavy outline are new relative to the mission beta baseline and represent significant design and implementation problems.

Table 4. Requirements-Functions Relationship

MISSION/SYSTEM REQUIREMENTS	KEY AVIONIC FUNCTIONS
• PASSIVE TF/TA • THREAT AVOIDANCE/TOLERANCE • IFFC/IFTC	• PROFILE OPTIMIZATION • INTEGRATED GUIDANCE/WEAPON DELIVERY
• SIMULTANEOUS ATTACK OF MULTIPLE TARGETS • AUTOMATION • CREW WORKLOAD RELIEF	• MISSION MANAGEMENT
• MULTISENSOR TARGET ACQUISITION • AUTOMATIC TARGET CLASSIFICATION • NIGHT/ADVERSE WEATHER	• IMAGE PROCESSING
• FAULT-TOLERANCE • RECONFIGURATION • POWER CONTROL • INTEGRATED MAINTENANCE	• SYSTEM FAULT DETECTION, ISOLATION, DATA ACQUISITION
• MULTIPLE DATA SOURCES	• DATA FUSION

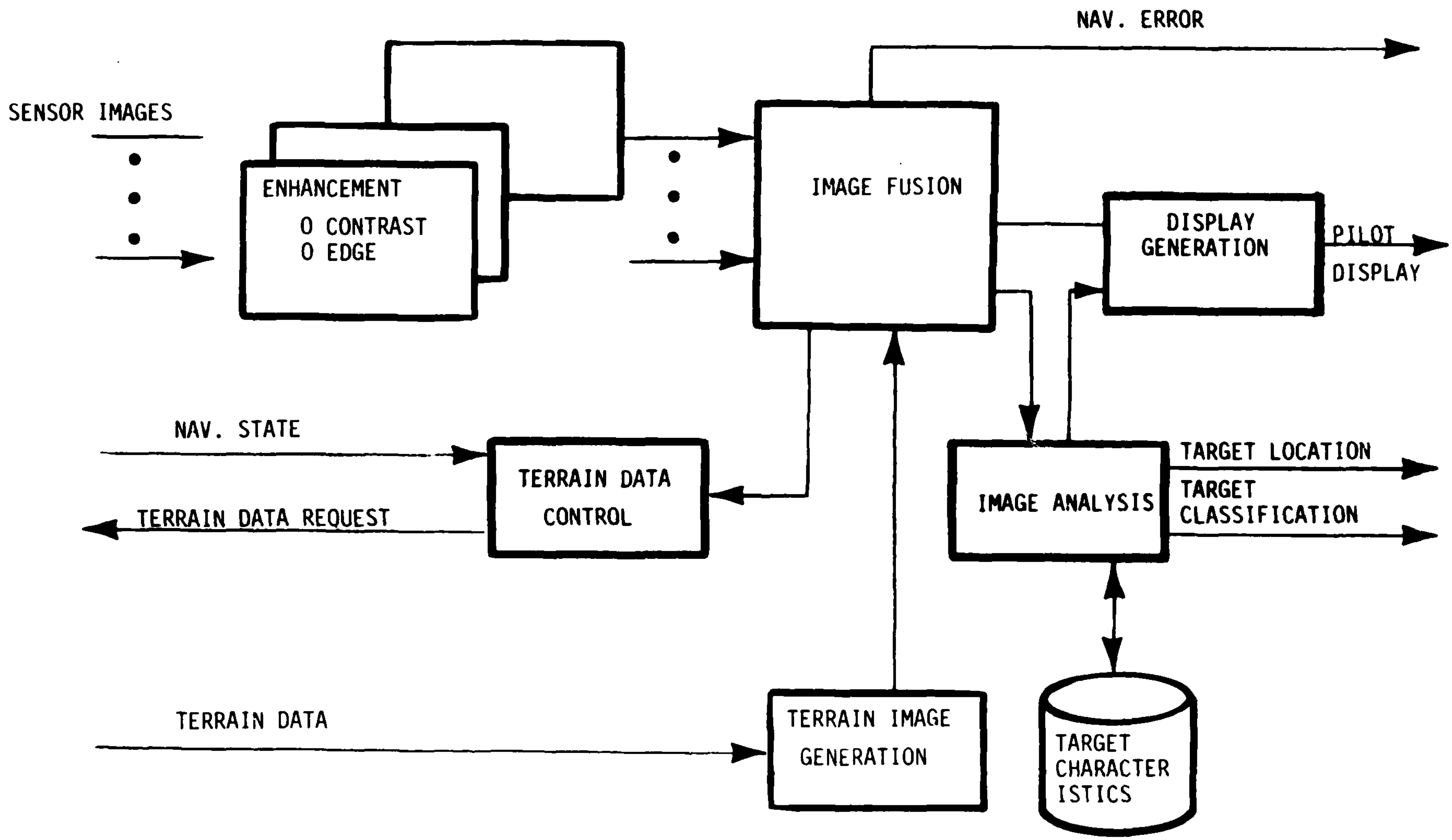


Figure 4. Image Processing

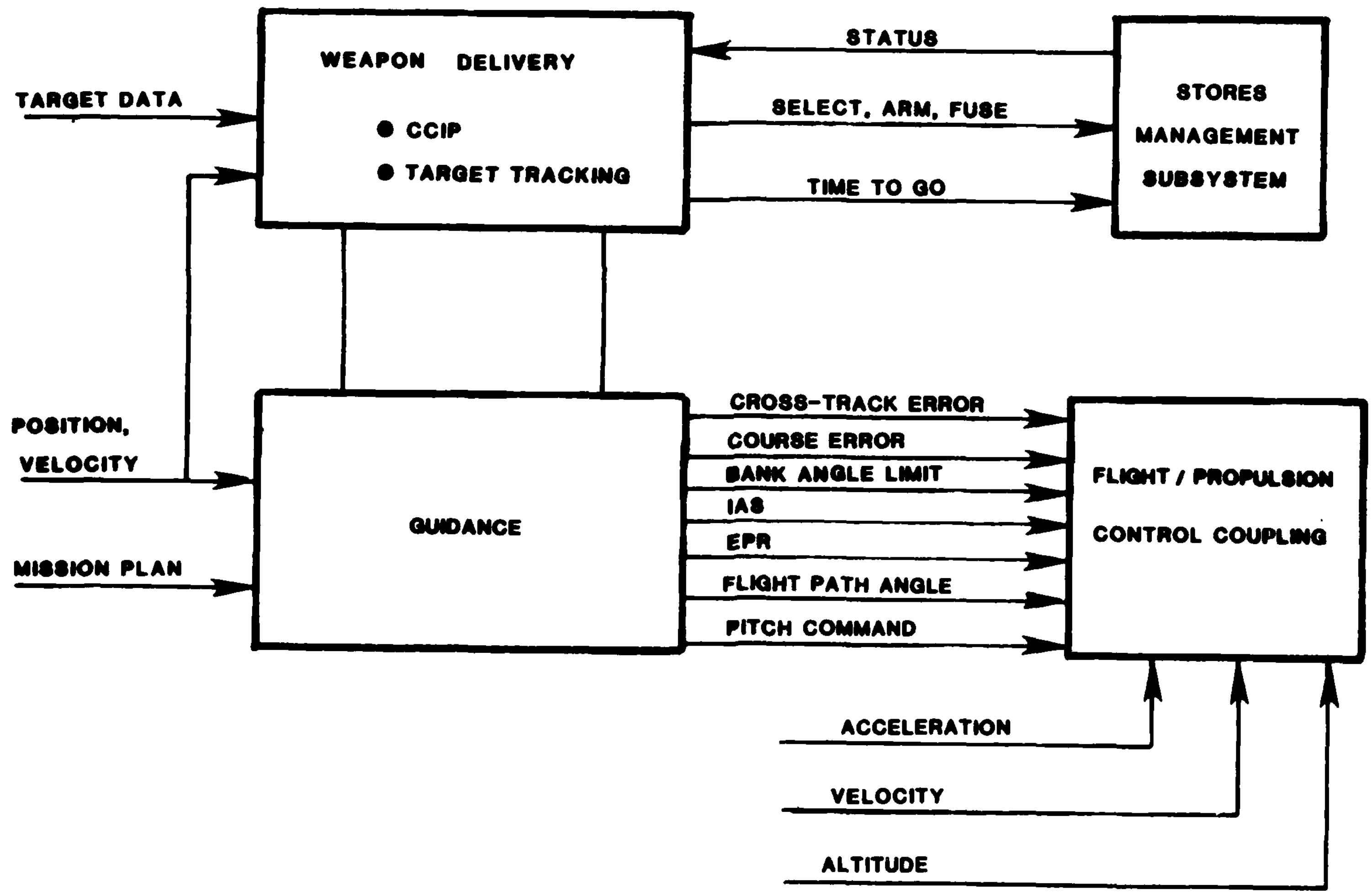
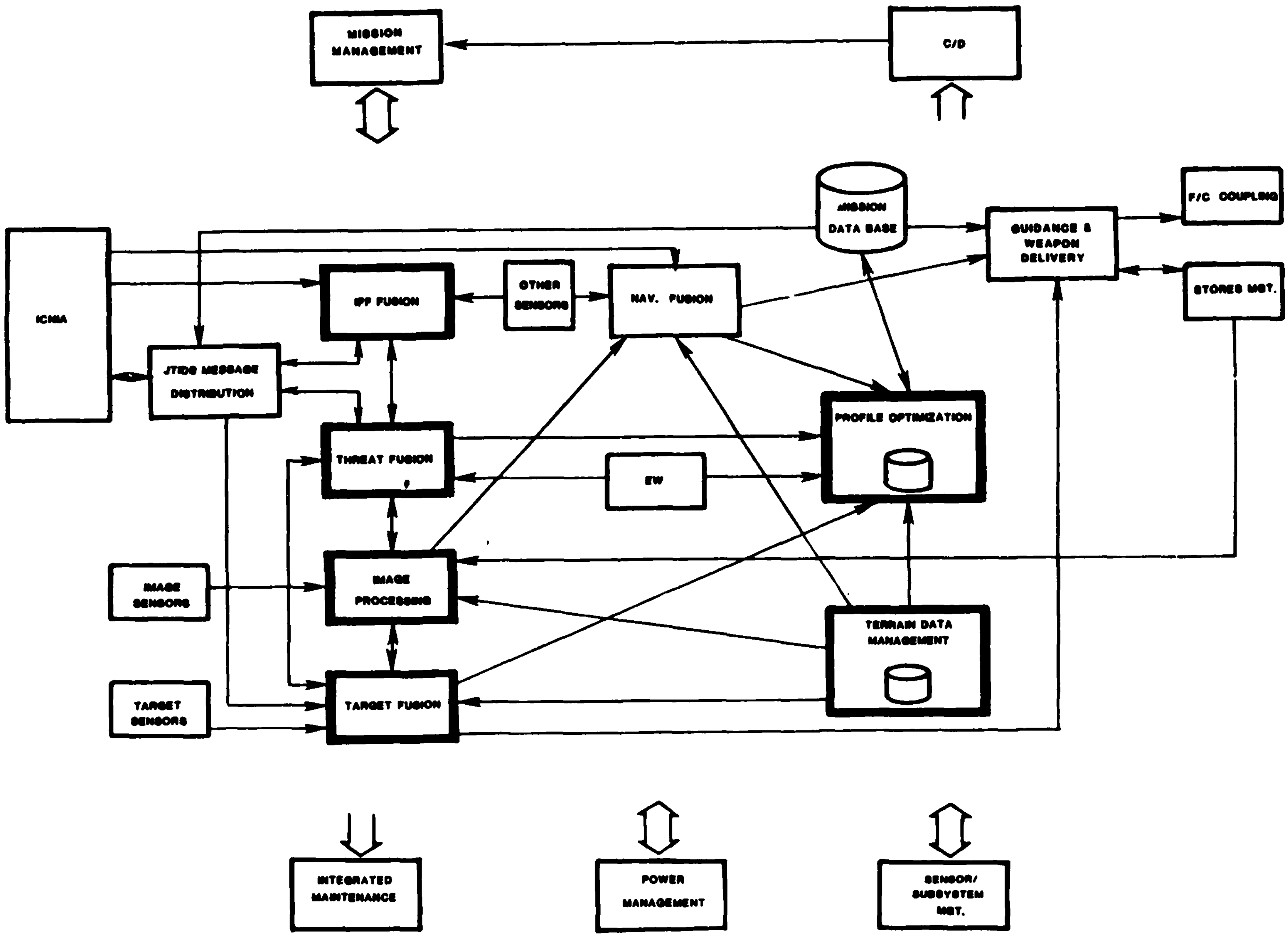


Figure 5. Integrated Guidance/Weapon Delivery


MISSION MANAGEMENT
C/D
MISSION DATA BASE
F/C COUPLING
GUIDANCE & WEAPON DELIVERY
STORES MGT.
ICNIA
IFF FUSION
OTHER SENSORS
NAV. FUSION
JTIDS MESSAGE DISTRIBUTION
PROFILE OPTIMIZATION
THREAT FUSION
EW
IMAGE SENSORS
IMAGE PROCESSING
TERRAIN DATA MANAGEMENT
TARGET SENSORS
TARGET FUSION
INTEGRATED MAINTENANCE
POWER MANAGEMENT
SENSOR/ SUBSYSTEM MGT.

3.2.4 Data Sizing And Flow Estimates

The terrain data will be accessed by multiple users (profile optimization, image processing, navigation, target fusion) simultaneously. Differing areas of interest and access pattern may be expected. As much as 400 megabits may be required to represent a 500nm by 500nm area.

With the above major subsystems defined the MAADS analysis then identified the function to function parameter flow. The results are represented on the "N-squared chart" shown in Figure 7. Major functions and subsystems are shown along the diagonal. Off-diagonal elements contain data flow estimates from one to another (in a clock-wise direction, e.g., the ij element is data from the ii to the jj elements). The letter designations are M, H or V for moderate, high or video data, respectively. The analysis showed the data communications grouped naturally into these three categories; a moderate speed, regular flow of about one to five megahertz for things like sensor data sampling; a high speed (20 MHz - 50 MHz) burst data flow for things like mass memory access; and video data for image related functions. The total indicate a 73.7 Kwords moderate data flow, a 37.2 Kwords high speed data flow and 18.2 Mwords for video data.

The central data processing throughput estimates made during the first phase analysis were then updated. In this exercise, profile optimization, image processing, and data fusion function were reserved to special purpose processors and the remaining tasks sized. The results are shown in Table 5. This preliminary sizing and timing estimate indicated three 0.5 MIPS processors would be required. A fourth was added as a monitor/backup.

The next sizing estimates were for the identified data bases. The results of this effort are shown in Table 6.

3.2.5 Strawman Definition

With these estimations completed, the MAADS effort was prepared to undertake the strawman architecture definition. Three 0.5 MIPS system processors would be required. A moderate rate data of about 70 Kwords per second must be supported. A high speed burst data flow averaging about 40 Kwords per second with an estimated peak of 1 to 3 Mwords per second must be supported. Video requirements of about 20 Mwords per second are to be expected. Major new functions included profile optimization, terrain data management, image processing, IFF fusion, threat fusion, target fusion, and system mass memory/data transfer module management.

IMAGE SENSORS						M 0.5		V8000			M 0.3					
	OTHERS					M 12.5	M 2.3	M 0.2				M 1.0				
		ICNIA					M 4.0					M 2.6				
			EW			M 0.4				M 12.0						
				STORES			M 0.3		OUTPUT DATA							
					C&D		M 0.2					INPUT DATA				
M 0.5	M 6.3		M 0.2		M 0.2	WEAPON SYSTEM MANAGE-MENT	M 0.2									
M 0.2	M 0.2	M 4.0	M 0.7	M 0.1		M 0.3	MISSION MANAGE-MENT	M 1.2	M 0.6	M 1.3	M 1.0	M 0.4	M 0.9	M 0.2		
					M 8.0 V 10200.		M 0.1	IMAGE PRO-CESSING		M 0.1	M 0.3					
							M 0.5		TERRAIN THREAT MASKING PROFILE					H 7.0		
						M 0.2	M 0.1	M 1.2	M 0.1	THREAT FUSION	M 0.2	M 1.1				
							M 1.3	M 2.5	M 1.0		TARGET FUSION					
		M 2.6								M 1.0		IFF FUSION				
													FLIGHT CONTROL			
							H 3.0 M 0.2	H 12.0	H 15.0		M 0.2			MASS MEMORY SUB-SYSTEM	TOTALS	
0.7	6.5	6.6	0.9	0.1	8.2	13.9	9.2	5.1	1.7	14.4	1.8	3.5	0.9	0.2	73.7	MODERATE S
							3.0	12.0	15.0		0.2			7.0	37.2	HIGH SPEE
					10200.0			8000							18200	VIDEO

Notes: All Data in K words per second, Data flow is clockwise.

Figure 7. Data Communication Requirements

Table 5. Central Data Processing Estimates

FUNCTIONS	MEMORY (KBYTES)	THROUGHPUT (MIDS)
EXECUTIVE	40N	.05N (N PROCESSORS)
POWER CONTROL	36	.075
FAULT DET./RECON.	30	.1
INT. MAINT.	25	.05
NAV FUSION	20	.05
PROFILE PLANNING	32	.25
F/C COUPLING	8	.02
GUIDANCE/WEAPON DELIVERY	144	.4
MISSION MANAGEMENT	100	.1
TOTALS	395 + 40N	1.05 + .05N

MEMORY	258K WORDS		THREE AYK-15A (~.5MIP)	PROCESSORS
	238K WORDS	OR	TWO .8 MIP PROCESSORS	
	218K WORDS		OR ONE 2MIP PROCESSORS	

NOTE: DOES NOT INCLUDE BACKUP PROCESSORS

Table 6. Data Base Summary

DATA BASE SUMMARY

DATA BASE NAME	SIZE (WORDS)	TYPE	SOURCE	MULTIPLE USERS	USERS (READ ONLY)
IFF TRACK	10K	UPDATE	UPDATED BY IFF FUSION	YES	IFF FUSION THREAT FUSION
TARGET STATE	2K	UPDATE	PRELOADED UPDATED BY TARGET FUSION	YES	IMAGE PROCESSING PROFILE OPTIMIZATION
TARGET/THREAT CHARACTERISTICS (CLASSIFIED)	5K	READ ONLY	PRELOADED	NO	IMAGE PROCESSING (ANALYSIS)
TERRAIN	25M	READ ONLY	PRELOADED	YES	TARGET FUSION IMAGE PROCESSING TERRAIN MASKING PROFILE OPTIMIZATION TERCOM

Table 6. Data Base Summary (Con't)

DATA BASE SUMMARY

DATA BASE NAME	SIZE (WORDS)	TYPE	SOURCE	MULTIPLE USERS	USERS (READ ONLY)
TERRAIN MASKED THREAT	100M	UPDATE (?)	PRELOADED UPDATED BY TERRAIN MASKING (?)	YES	PROFILE OPTIMIZATION IMAGE PROCESSING
MISSION PLAN (CLASSIFIED)	20K	UPDATE	PRELOADED UPDATED BY PROFILE OPTI-MIZATION (?)	YES	EW (ECM) PROFILE OPTIMIZATION
INTEGRATED MAINTENANCE	20M (1% DATA ON 4 HR MISSION)	WRITE ONLY	UPDATED BY INTEGRATED MAINTENANCE	NO	
NAV AIDS	10K	READ ONLY	PRELOADED	NO	ICNIA
THREAT I.D. (CLASSIFIED)	5K	READ	PRELOADED	NO	EW

It was assumed the system processors could be 1750A type computers with multiple channels, intelligent 1553B channel interfaces and capable of 0.5 MIPS or more throughput. The bus technology available was assumed to include at least one 1553B bus which for sizing purposes was rated at an effective 30 Kwords per second capacity. For the moderate data flow MAADS postulated a "fast 1553" or perhaps multiple 1553s to support the 70K words(which implies a bus rate of 3 to 5 MHz). High speed bus rates were taken to be in the 20 MHz to 50 MHz range based on the kinds of discussions and consideration being undertaken by the high speed bus ("A2K") committee and hence a single bus can easily handle the projected load. Video buses were postulated to be in the 300 MHz to 500 MHz range.

Of the data bases identified, the terrain data, the terrain masked threat data and the mission plan were viewed as serious candidates for mass memory storage, though not necessarily a single or even the same type of device. The mass memories will clearly require both 1553 and high speed bus interfaces. Other complications include the impacts of multi-user access, the influence of classified data bases and the loading of mission peculiar data bases.

The influence of fault tolerance on the strawman architecture was multi-faceted. It was assumed fault tolerance is addressed in the design of each subsystem. System functional redundancy and system level fault detection and isolation will be used to improve system fault tolerance. A spare processor would be employed. Software must be able to tolerate data from malfunctioning subsystems and be adaptable to a changing configuration.

The partitioning of tasks between the system and subsystems involved several concerns. A degree of functional modularity is desirable to enable the configuration of an aircraft to a variety of mission requirements (it is unlikely that economics will permit all aircraft to carry all functions, particularly the advanced and specialized functions). The need for special purpose processing precludes total resource sharing and suggests the distribution of such processing to the subsystem. Improper partitioning could adversely affect system performance.

The result of all these considerations was the strawman architecture as shown in Figure 8.

The following features of the strawman configuration should be noted:

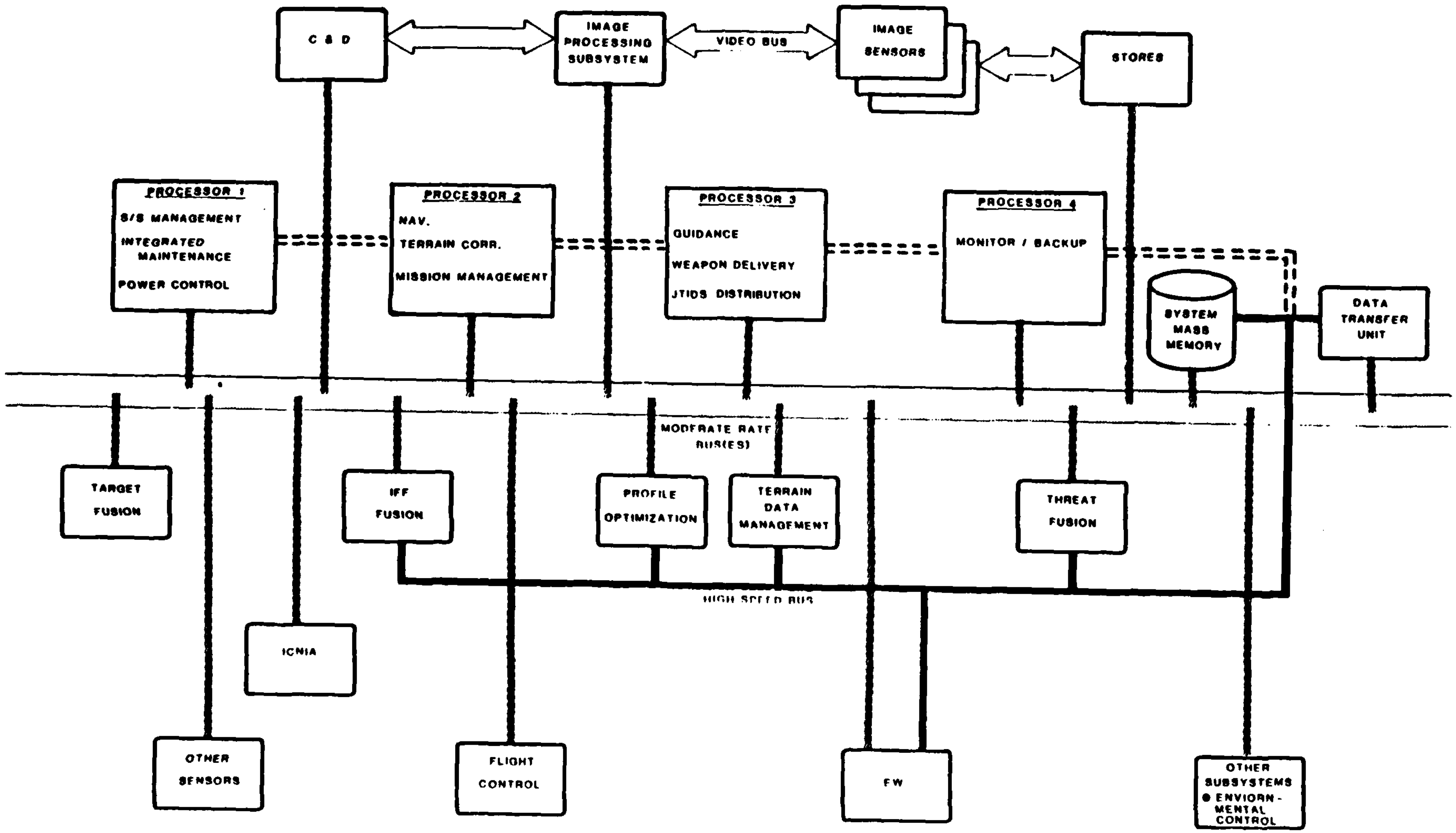


Figure 8. A STRAWMAN System Configuration

- a moderate speed global bus composed of three 1553B buses or one 1553B plus a higher speed 1553 bus

- a regional high speed "data burst" bus which provides special mass memory access for selected subsystems (may extend to processors in a future configuraton)

- a dedicated video bus for simultaneously transmitting multiple images between sensors, image processing and displays

- a Data Transfer Unit for on-loading large mission peculiar data bases (e.g., mission plan, threat location file, etc.) and off-loading maintenance data for post-flight processing

- four central processors, with a preliminary partitioning of major functions

- various "smart" subsystems containing special purpose processors as well as some conventional data processors.

- miscellaneous other sensors

The strawman configuration is flexible in a number of specific implementation details. The number of processors or buses could be changed without architectural impact. No specific control strategy is implied. Subsystem configurations could be adjusted. It is adaptable for other functional partitioning or new technology.

Note in particular that a different partitioning of the functions tentatively grouped in the Advanced Digital Avionics Map (ADAM) subsystem is suggested based on the following reasoning:

- profile optimization and image generation are distinctly different technologies and would not both be required by all user aircraft.

- images derived from terrain data would be useful to the image processing functions--inclusion of the terrain image generation in an image processing subsystem would minimize bus traffic.

- terrain data is required by several users and is mission dependent unless a very large data base can be accommodated by the mass memory. A data management function is required and should be a system level function.

- A TERCOM function requires only modest processing and local data base update. Inclusion in the system level navigation function may be the most economical use of processing resources.

The system architecture can evolve with technology improvements as shown in Table 7. A future system will likely have VHSIC processors with substantially higher throughput, may have generic signal processing elements as shared system resources and could have a global bus capable of several kinds of data transmission, including 1553B. The system pictured

Table 7. Architecture Evolution

	NEAR TERM	MID TERM	FUTURE
GP PROCESSOR	ENHANCED 15A	VHSIC	
SIGNAL PROCESSOR	DEDICATED, SPECIAL PURPOSE	LIMITED GENERALIZATION, MORE SHARED RESOURCES	GENERIC SIGNAL PROCESSING ELEMENTS
BUSES	• GLOBAL 1553B (THREE) • REGIONAL HIGH SPEED BUS • DEDICATED VIDEO BUS	• GLOBAL 1553B AND HIGH SPEED BUS • DEDICATED VIDEO BUS	• GLOBAL MULTIMODE, MULTICHANNEL, SINGLE PHYSICAL CARRIER
EXEC	ENHANCED DESCENDANT OF DIALS EXECUTIVE	DUAL-MODE EXECUTIVE (1533B AND HIGH SPEED)	MULTI-MODE
SUBSYSTEM PARTITIONING	BOXES DEDICATED FUNCTIONS	BOXES DETERMINED BY COMMON TECHNOLOGY	

in Figure 9 includes these features and illustrates a subsystem partitioning based on technology rather than function. The advantage of the latter approach is to maximize the sharing of common resources, and provide the greatest degree of flexibility in system configuration. It exhibits the highest degree of functional integration and a maximum modularity resulting in minimum spares provisioning.

3.2.6 System Control Issues

In parallel with the definition of the strawman configuration, the MAADS effort began to identify issues related to control strategies. There are a number of issues and alternatives that need to be investigated. The outcome of these investigations will ultimately be the System Control Procedures and the preliminary Executive design.

Building on the DAIS system control procedures and using the strawman configuration for guidance the following list of system control topics was generated:

- o System Startup and Initialization
  - o Pilot Interface
  - o Power Control
  - o Initial System Load
  - o System and Subsystem Initialization

- o Shutdown
  - c Pilot Interface
  - o Power Control

- o System Restart
  - o Pilot Interface
  - o Pilot Initiated Restart
  - o Recovery Restart
    - Power Transient (EMP, Nuclear)
  - o In-Flight Reload

- o Normal System Operations
  - o Definition of "High Speed Bus"
    - Characteristics
    - Protocol
    - Processor Interface
    - Control Procedures
  - o Centralized or Distributed Bus Control
  - o Centralized or Distributed Time Synchronization
    - Task Execution
    - Data Flow
  - o Time-Stamping for Time Critical Data
    - Time Resolution
    - Real-Time Clock
      - Processors
      - Subsystems
    - Time Resolution Compatibility

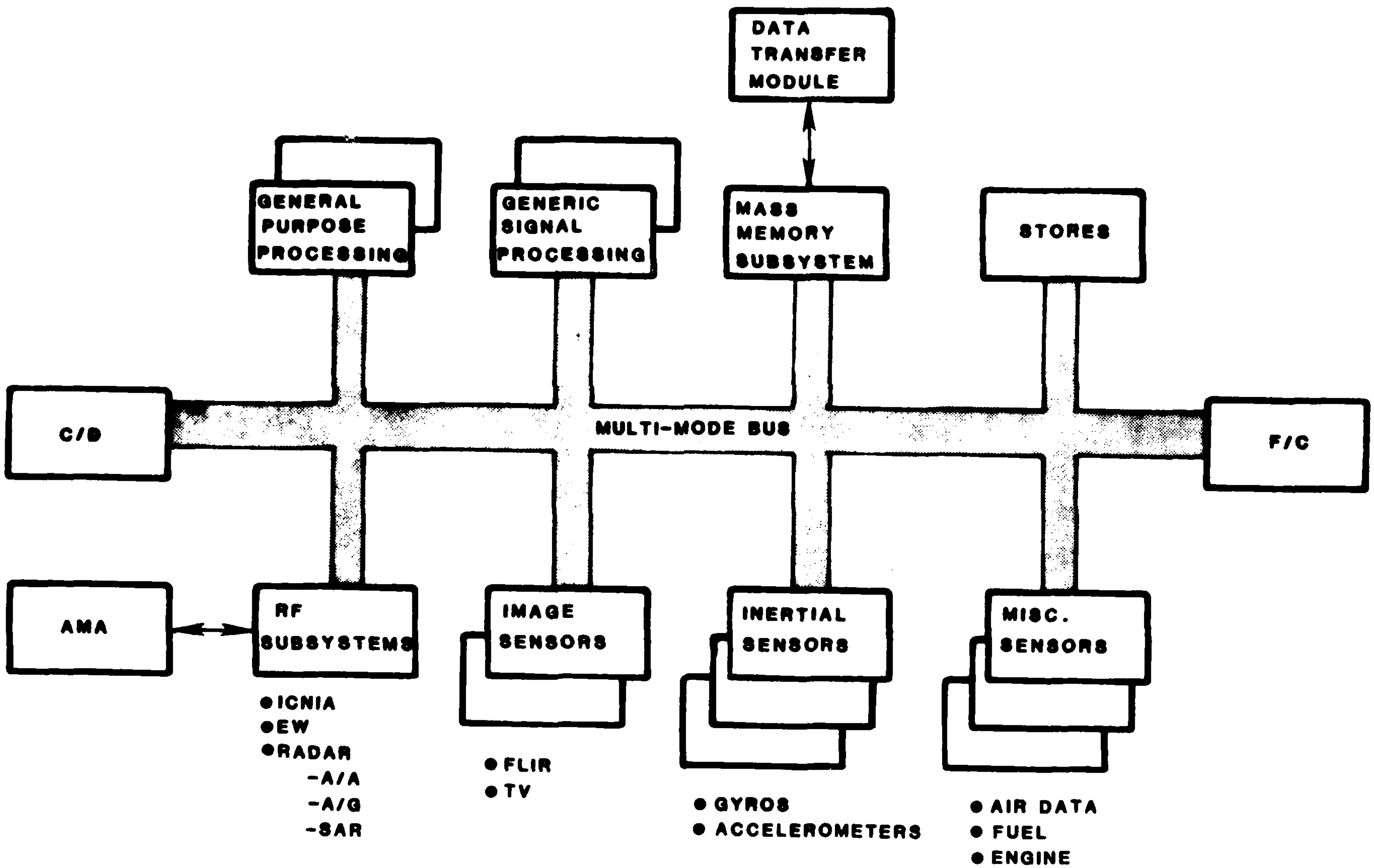


Figure 9. Future Architecture (SMOKEMAN)

- o Mass Memory Interface
  - 1553B Interface
  - PIO/DMA Interface
  - High Speed Bus Interface
- o Message Types
  - Minor Cycle Synchronization
  - Synchronous Messages
  - Asynchronous Messages
    - Critically-Timed Messages
    - Background Messages
      - Protocol
    - Control Messages
      - Definition
      - Protocol
    - High Speed "Burst" Data
      - Definition
      - Protocol
      - Control
  - Special Messages
    - Idle Polling

- o Detect Errors and Recover
  - o Software Errors
  - o Data Errors
  - o Retry Philosophy
    - Eliminate Sensitive Data Retry
  - o Error/Exception Processing Definitions
    - Error Logging
    - Terminal Flags, Subsystem Flag Logging

- o Resource Management
  - o Reconfiguration
    - Backup/Redundancy
    - In-Flight Reload
  - o Device Failure Detection
  - o Device Recovery

- o Emergency
  - o Emergency Take-Off
  - o Classified Erase

- o Define Executive to Application Software ICD

- o Define Bus Control Interface to Processor ICD
  - o 1553B
  - o High Speed

- o Define Executive Data Base Generator Software
  - o User's Interface
  - o Executive Interface
  - o Functional Requirements
    - Validity Checking
    - Operational Analysis

Some of the issues identified are issues beyond the scope of what AFWAL programs have addressed to date. Other issues have been addressed and approaches adopted on previous programs.

Of course, the new issues require new approaches. The issues addressed on previous programs fall into several categories:

1. some require new approaches because of additional requirements

2. some can be approached with new methods because of technological improvements

3. others have been successfully solved and nothing can be gained by changing

For example, in the new issues category are power control during startup and shutdown, definition of the "high speed" bus, definition/protocol of the "burst" data, and the emergency procedures.

Definition of the mass memory interfaces is an example of the categories for which a new approach is required. The DAIS approach to mass memory will simply not handle the requirements of PAVE PILLAR.

Some areas that can be improved because of the new topology and technology advances include:

- centralized or distributed bus control
- centralized or distributed time synchronization
- error logging/terminal flag, subsystem flag logging
- device failure, recover
- initial loading, in-flight reload
- power transient recovery

3.2.7 Mass Memory Issues

Although listed under the system control topics, the mass memory issues were also singled out for individualized consideration. This is because a critical area of concern in future avionics is the increasingly large requirement for mass memory storage. MAADS identified the principal issues to be addressed:

- Interfaces

- o Multi-User Access
- o File/Format Standardization
- o Ground Support Compatability
- o Segregated/Removable/Erasable Classified Data
- o Technology - types of memory

Three types of interfaces are possible and all three must be defined including a record-oriented 1553 interface as opposed to a word-addressable interface. File/format definitions is a very important area of future standardization especially in light of ground support requirements implied by mission data being loadable on the flight line. The MAADS/PAVE PILLAR effort must carefully track new mass memory technology.

The use of a data transfer unit (DTU) to accommodate "carry on" data bases is identified as an important feature of future systems. This will support rapid flight line loading of mission data, for example mission plans and new threat intelligence.

The DTU is not a new concept but could be enhanced by new technology. "Carry off" data may be increasingly important in future systems for maintenance data, trend analysis, and threat updates.

3.2.8 High Speed Bus Considerations

The area of the high speed data bus is another singled out for individualized attention. the MAADS effort observed that the requirement to rapidly transfer large blocks of data asynchronously is incompatible with the regular, synchronous, data flow/processing of many avionics subsystems/algorithms. Such data should be transmitted over a separate high speed channel even though the total data rate is not much beyond the capacity of 1553. The standardization of high speed buses is the subject of considerable activity, exemplified by the SAE subcommittee on high speed data buses (known as the A2K committee and later upgraded to the AE-9B committee).

The MAADS effort actually found little in the way of firm requirements for high speed buses. But, it was felt that this was an area where requirements will materialize rapidly as the technology appears imminent.

An important distinction is made between the basic hardware technology of high speed buses and the protocol or methodology of implementing a communication channel on the basic hardware. The MAADS effort identified the protocol and its potential standardization as the area requiring the most attention. Two candidate protocols were thought to be under active consideration by the A2K committee. There were ETHERNET, a 10 megabit contention/collision detection type of approach and "FREE ACCESS", a 50 megabit proposal featuring contention resolution prior to message transfer.

These two protocols were identified as candidates for more analysis in the next phase of the MAADS effort. A preliminary position was reported to the Air Force that MAADS did not expect ETHERNET to be suitable for avionics, whereas FREE ACCESS appeared to be a good starting point for a standards definition effort. A general caution was also raised at the System Assessment Review that much of the technical literature on the subject of communication networks actually addresses the general purpose, commercial network which, in several key aspects, is a substantially different problem than an avionics data bus.

3.2.9 SRR/SAR Comparison

Near the conclusion of the System Assessment Review a comparison was drawn between the results of the first two phases of the MAADS effort. The architectural approach defined in the first phase was compared to the strawman configuration. The two 1553B buses and the high speed 1553 are represented in the STRAWMAN configuration as moderate rate bus(es). As to the exact number and mix of buses to support this moderate data flow, MAADS considered the architecture and executive insensitive to this question and therefore did not specify them for the STRAWMAN configuration.

It was also noted that the high speed bus, interconnecting subsystems and not attaching to avionics processors, has become, in the strawman configuration, two buses. One is a very high speed, video bus with the same connectivity (to subsystems, not processors) and the other is a high speed digital data bus which connects to the mass memory and potentially to an advanced avionics processor. The point was also made that complex subsystems with hierarchial structures still persist in the STRAWMAN configuration, though not sketched in detail. Also, the STRAWMAN should not be misinterpreted as requiring all mass memory to be at the system level; additional mass memory may be dedicated to functions and reside in subsystems.

3.3 Interim Technical Review

3.3.1 Overview

The third phase of the MAADS effort lead up to the Interim Technical Review and expanded the level of detail sufficient to complete several specifications. These included the Mission/Environment Specification, the System Requirements Specification, the System Architecture Specification, and a Computer Program Development Specification for avionics real time operating system software. This period of the study also included an in-depth analysis and comparison of protocols for the high speed bus. An informal report fully documenting this analysis was also delivered at the time of the Interim Technical Review. This report is included here as Appendix A.

3.3.2 Mission/Environment Specification

The Mission/Environment Specification describes the tactical missions projected for the 1990s and the conditions under which they are to be accomplished. This represents one of the primary outputs of the MAADS effort and is to be used as the baseline for the Avionics System Integration Demonstrator (ASID) System Definition project. The Mission/Environment Specification describes the typical air-to-ground missions and the actual theaters of operations. It includes existing and projected correlated threats, correlated weather, actual targets, advanced weapons and describes representative operations. The significant new functions defined are path optimization, integrated path control/weapon delivery, integrated test and maintenance, data base management, image processing and dynamic resource allocation.

The tactical missions covered were close air support (CAS), battlefield air interdiction (BAI), air interdiction (AI) and defense suppression (DS). The missions were described in terms of the mission objectives, the types of targets, target locations, and the type of coordination required. Day, night, and all weather conditions were specified. The representative theaters specified were Europe, the Middle East, and Korea. This provided a diversity of terrain, weather, threats and EW complements for the missions. The weapons complement specified included those currently available, those in development, and future potential technologies.

3.3.3 System Requirements Specification

The System Requirements Specification provides the overall system definition and describes the processing characteristics in terms of the information management, mission management, and system management. A high level representation of this is shown in Figure 10. Mission management includes command processing, display management, mission planning/profile optimization, and system/mission monitor functions. System management encompasses vehicle path control, sensor/subsystem management, EW management, CNI processing, integrated test and maintenance functions, and power system management.

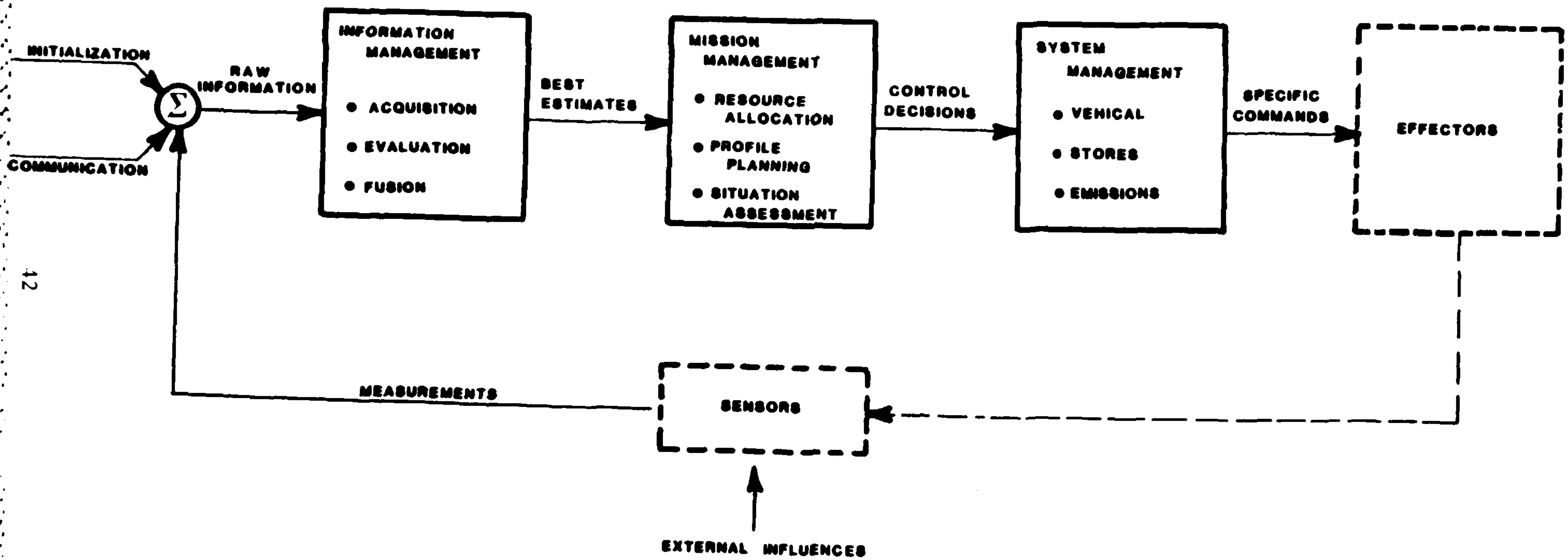


Figure 10. Avionics System

3.3.4 System Architecture Specification

The System Architecture Specification defines the system elements or components out of which avionics systems for future tactical aircraft will be built. It also defines the topology of such systems and the control strategy and used to manage the systems. In addition to these three categories of components, topology, and control strategies, the MAADS effort decided to also explicitly call out standard interfaces as a key aspect of the architecture. While interfaces can be considered as part of the topology definition it was felt that standard interfaces are of such critical importance to future avionics systems that separate discussion of them was justified.

3.3.4.1 Hardware Interfaces

Hardware elements defined include computational elements, addressable memory, a standard backplane interconnect, system serial data paths, sensor interface modules and mass storage. Computational elements consist of MIL-STD-1750A processors (both LSI and VHSIC) and MIL-STD-1862 processors. Memory elements include both non-volatile (core, bipolar PROM, etc.) and volatile (static RAM, VHSIC SRAM) with word parity a minimum requirement. Cache memories are specified to compensate for speed mismatches. A connection to the standard backplane interconnect is specified. The standard backplane is specified to provide a standard interface for the other elements in system nodes. The IEEE-P896 effort now underway is identified as the source of a potential standard. System serial data paths include MIL-STD-1553B buses and a high speed bus. Intelligent bus control interfaces are specified for both these serial data paths. Standard sensor interface modules are specified. These include analog-to-digital, digital-to-analog, discrete in/out (TTL), discrete in (contact closure), discrete out (high current output), synchro-to-digital, and digital-to-synchro. An I/O register for status and control is specified as well as programmability via a DMA capability and a programmable interrupt capability. Mass storage is specified for both global and regional applications. The Shugart Associates System Interface (SASI) bus is recommended as a baseline standard with host adaptors and control units resident with the memory devices. Candidates for future standard elements were identified to include: generic signal/array processors, video buses and interface units, display generation/image processing elements, high speed parallel buses for intranode/internode transfers and multi-port memories for data buffering between elements within a node. A list of hardware standard interfaces and their current status is shown in Table 8.

Table 8. Standard Hardware Interfaces

INTERFACE	STATUS	
	FUNCTIONAL	ELECTRICAL/ MECHANICAL
MIL-STD-1553B (TO BUS)	MIL-STD	MIL-STD
HIGH SPEED BUS (TO BUS)	DEFINED	DEFINED
MIL-STD-1553B CONTROLLER TO PROCESSOR	DEFINED	DEFINED (SBI)
HIGH SPEED BUS CONTROLLER TO PROCESSOR	DEFINED	DEFINED (SBI)
STANDARD BACKPLANE INTERCONNECT TO OTHER ELEMENTS	CANDIDATE IDENTIFIED	CANDIDATE IDENTIFIED
SENSOR INTERFACE MODULE TO PROCESSOR	NOT DEFINED	DEFINED (SBI)
SENSOR INTERFACE MODULE TO SENSORS/ACTUATORS	NOT DEFINED	DEFINED (SBI)
MASS MEMORY TO PROCESSOR	NOT DEFINED	DEFINED (SBI)
AVIONICS REAL-TIME OPERATING SYSTEM TO APPLICATIONS	DEFINED	N/A

3.3.4.2 Software Interfaces

In addition to the hardware elements, standardization of some software elements is recommended. Software is viewed as a layered structure consisting of a nucleus which may be identical for all systems and system nodes, a system control layer which would be configurable for each system and/or system node and an applications layer which would be system design dependent and/or mission dependent. Standardization of the nucleus which performs task dispatching, event handling, I/O interface handling and the like is recommended. Standardization of the system control software which provides extended machine support, bus control, mass memory control, sensor interface control, etc., is also recommended although the configuration of these system control modules would remain flexible.

Other software elements consist of support tools which should be standardized. These include a high order language (HOL) compiler, a 1750A assembler, a linker (which must support the expanded memory capability) and a system generation tool such as an advanced version of the Partitioning and Linkage Editing Facility (PALEFAC) currently used in the Avionics laboratory.

3.3.5 Design Example

At the interim technical review the MAADS effort also displayed a design example to show how one implementation of the recommended architecture might appear. This sample is shown in Figure 11 (hardware) and Figure 12 (software). As part of this design example a software sizing exercise was performed for the indicated functions. The result of this exercise is presented in Table 9. The control strategy selected for this design example was a round robin approach implemented via the dynamic bus control mode command. This was augmented with a bus monitor/timeout function in each potential controller to recover from sudden failure of the current controller. Hot backups were specified for critical functions to facilitate rapid reconfiguration/recovery.

3.3.6 High Speed Bus Analysis

The interim technical review also presented an in-depth review of the analysis conducted in the area of high speed buses. This first involved the identification of the desirable attributes of a high speed bus. These included fault tolerance, efficiency, simplicity, assurance of data integrity, support of synchronous systems, adaptability to new technology, similarity to MIL-STD-1553B and deterministic characteristics.

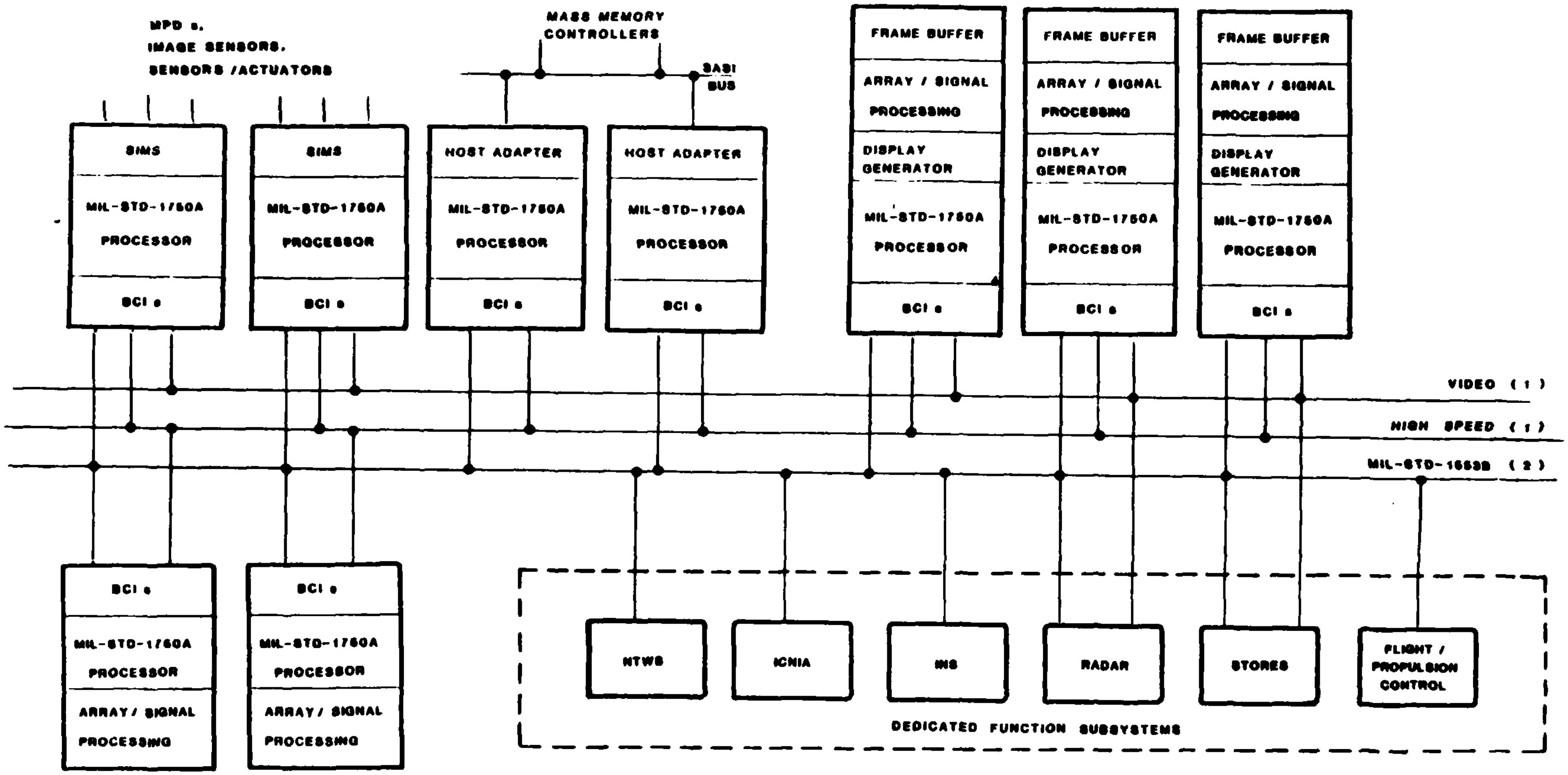
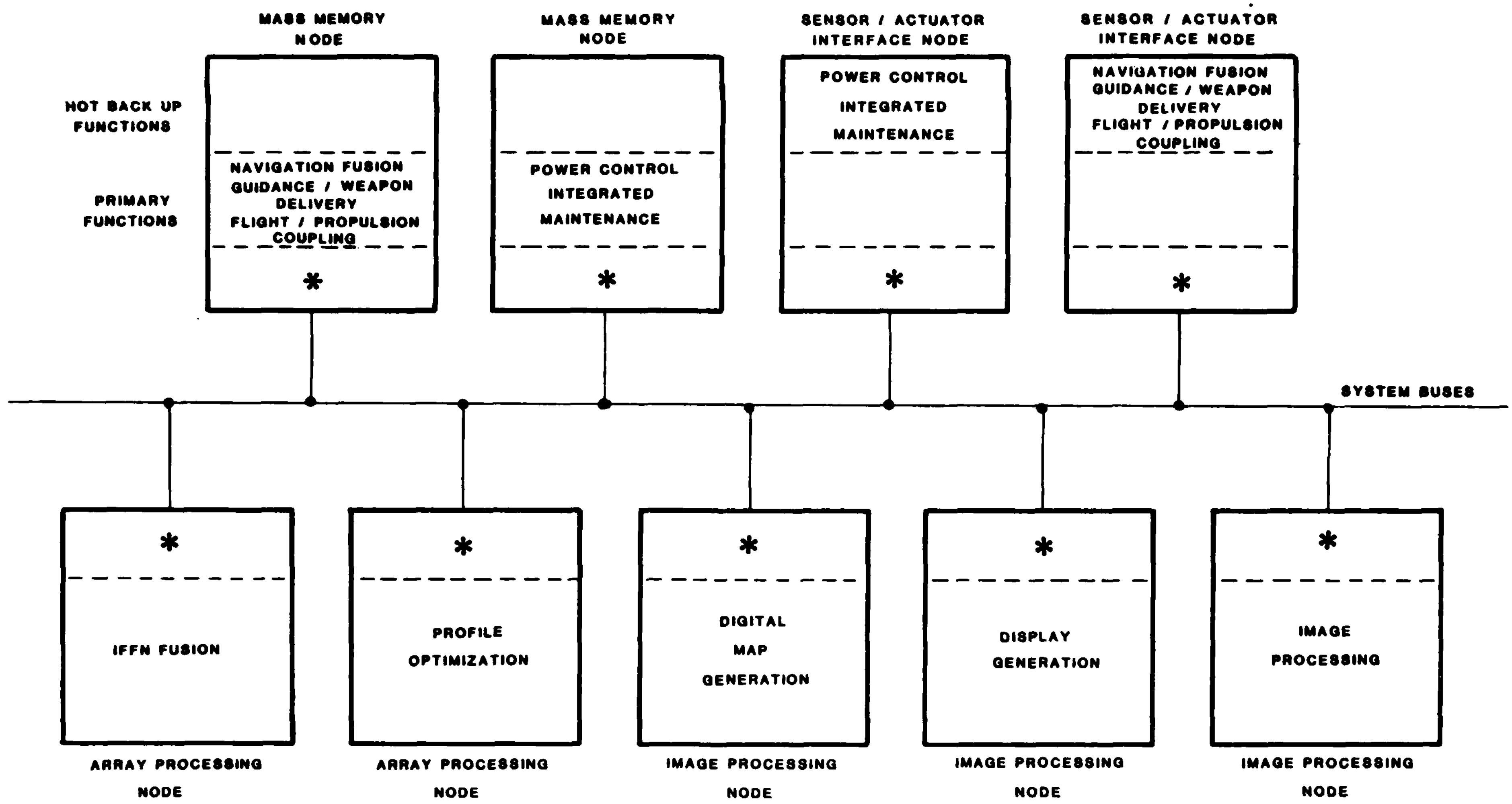


Figure 11. Hardware Configuration Design Example



* OPERATING SYSTEM, FAULT DETECTION/RECONFIGURATION , AND MISSION MANAGEMENT SOFTWARE

Figure 12. Software Configuration Design Example

Table 9. Software Sizing Design Example

FUNCTION	MEMORY (KWORDS)	THROUGHPUT (MIPs)	NODE TYPE
Operating system	30N*	.05N*	All
Fault detection/reconfiguration	5N*	.05N*	All
Mission management	10N*	.05N*	All
Integrated maintenance	12.5	.05N	General Purpose Processing
Navigation fusion	10	.05	General Purpose Processing
Flight/Propulsion coupling	4	.02	General Purpose Processing
Guidance/Weapon delivery	72	.4	General Purpose Processing
Power Control	18	.075	General Purpose Processing
Image processing (coding, restoration, enhancement, feature extraction)	Unknown	Unknown	Image Processing
Digital map generation	Unknown	Unknown	Image Processing
Display generation	50	.5	Image Processing
IFFN fusion	Unknown	Unknown	Array Processing
Profile optimization	Unknown	Unknown	Array Processing

*Distributed function resident in each mode.

The generic protocols analyzed and evaluated consisted of collision detection, time slots, token passing, round robin, priority polling, priority overlay, priority timeout, and free access. Of these, the round robin, priority polling, and free access approaches were judged sufficiently promising to be subjected to a comparative throughput analysis. Assumptions and parameter selections necessary to accomplish this analysis were reviewed in detail. The informal report documenting the analysis performed is included in this report as Appendix A. (Appendix B documents a more formalized protocol evaluation performed after the interim technical review upon specific Air Force direction.)

The results of the throughput analysis showed that the free access approach was least effective and it was therefore dropped from further consideration. It was then observed that the priority polling and round robin approaches were not mutually exclusive, i.e., either could be designed within the context of a single protocol. Next, it was observed that the characteristics of the throughput equation indicated that a stationary master approach should do well. When computations for this case were performed, the stationary master proved to be up to twice as effective as the other approaches. Also, it was recognized that the stationary master is not contradictory to the other approaches. In fact, the round robin and priority polling both operate in the same fashion as the stationary master for at least the duration of an individual message. The protocol adopted therefore specifies the inclusion of the dynamic bus control mode command and leaves the system designer the freedom to select how it will be used. The protocol supports a round robin, priority polling, or stationary master control strategy, or some hybrid of the three as appropriate for a specific system design. In general, the stationary master provides the best throughput; round robin provides the quickest control transfer; priority polling provides the most predictable response for an emergency message.

3.3.7 Bus Control Interfaces

The interim technical review then recapped the status of the specified I/O controllers. Specifications were delivered for a 1553B bus control interface (BCI) and a BCI for a high speed bus. A mass memory interface is defined with a candidate baseline identified. The need for a BCI for the video bus is identified but this represents a new development yet to be undertaken. The defined BCIs for the 1553B bus and the high speed bus are intelligent controllers. They are programmable with MIL-STD-1750A type instructions, using the same format and operation codes wherever possible. The interfaces are controlled from the processors via a high level of XIO commands. they implement exception chains that function in place of processor interrupt handlers, performing processing based on BCI generated error/exception/completion codes. On board 8K RAM memory is specified for storing instructions, data pointer tables and exception chains. Multimode operations are specified including master, remote and monitor modes. Both the BCIs have a common interface to the MIL-STD-1750A CPU. Both are compatible with a flexible protocol that accommodates a round robin, priority polling or stationary master control strategy. Both are designed to operate in a "start and forget" fashion as far as the processor is concerned.

3.3.8 Operating System Software

The software operating system is specified based on a modified executive concept that separates mission independent software into that which is also design independent and that which is design dependent. The nucleus incorporates those functions that are independent of both mission and design. The system control tasks (SCT) address functions that are independent of mission requirements but dependent on design approach. This will result in increased flexibility and growth potential while yielding a smaller operating system in each processor.

The nucleus provides task and event handling, manages I/O interfaces and supervises local processing. System control tasks provide extended machine support and administer control over buses, mass memory, sensor interface modules, display and other special purpose processors. This will permit many different approaches to distributed processing without an entirely new executive for each implementation.

The avionics real time operating system (ARTOS) to application software interface control definition covers both declaration statements and real time statements. Declaration statements include task declarations, event declarations, queue declarations, data block declarations, file declarations and common subroutine declarations. Task/event management is accomplished via statements such as a schedule, cancel, terminate wait (on time), wait (on event) and signal. Data management is achieved via statements such as read, write, send, receive, file read, file write and trigger. I/O modification is via statements SIL add and SIL delete. Terminal status statements include fail terminal, recover terminal, and attempt recovery. Real time built in functions consist of event read, task execution status read, mission time read, minor cycle read, terminal data field read and clear terminal field. With these facilities provided, application software may be written to concentrate on the avionics algorithm being implemented and not be burdened with the routine support function that must be incorporated in any system.

3.3.9 System Control Procedures

The final topic covered in the interim technical review was the system control procedures as documented in the architecture specification. These procedures cover system time, process control, communications protocol, communication error handling, device failure, startup and shutdown, system restart, in-flight reload and processor backup.

3.4 Implementation Of Additional Functions

The results of the studies and trade-offs indicated that many of the additional functions could be supported by modification of existing AFWAL/AVSAIL software. The modification of this software, however, required the use of support tools. Since these tools were only available on a DEC-10 and the DEC-10 would not be available in the future, the support tools needed to be rehosted to a machine that would support future development. The host selected was a VAX. The AFWAL/AVSAIL executive software was then rehosted from the DEC-10 to a VAX 11/780.

4.0 CONCLUSIONS AND RECOMMENDATIONS

The Multibus Avionic Architecture Design Study (MAADS) has successfully reviewed and evaluated projected avionic requirements for tactical aircraft of the 1990s and defined an architectural approach and a design example suitable for use as the baseline for the Avionic system Integration Demonstrator (ASID) System Definition efforts.

The architectural approach is indeed multi-bus in nature, having the appearance of a parallel bus structure at the highest system level and taking on the characteristics of a hierarchical structure at level of functionally isolated subsystems. The approach is highly flexible, allowing future system designs to exploit the best features of both structures in accord with the needs of a specific mission and system. The approach will accommodate the infusion of new technologies and, in fact, accelerate this process by providing a context in which system designer may focus on the true technical issues associated with new technologies.

The significant achievement of the MAADS effort is the definition of an architectural approach which at one and the same time encourages the movement toward increased standardization of system elements and interfaces (with the acknowledged benefits thereof) and yet increases the design freedom available to system implementations. The multibuses defined include the already standard MIL-STD-1553B, a high speed bus which should be standardized as soon as practical and a video bus which needs more analysis and definition.

The MAADS effort has successfully built on the DAIS architecture, maintaining desirable characteristics such as system symmetry from the viewpoint of the core general purpose processors while providing mechanisms to overcome shortcomings such as dealing with the conflict between synchronous and asynchronous operations.

MAADS has identified the key near-term technology development need to be that of multi-channel, smart-channel interfaces for MIL-STD-1750A processors. This technology should also be infused into the VHSIC program as quickly as possible.

MAADS has identified the need to move standardization down another level to the in-box components such as processors, memory modules, and I/O interfaces. ICNIA and INEWS are seen as developments that have already initiated this new thrust in standardization.

The standardization of software elements as well as hardware elements is also recommended. An avionic real time operating system (ARTOS) is defined consisting of a nucleus and a layer of system control tasks. Like the hardware standardization, this will accelerate and enhance future development by removing the repetitive and burdensome developments required in all systems and thus allowing concentration on the specific avionic algorithms of interest. The interface standard proposed will provide the mechanisms to accomplish the task without restricting design freedom.

The MAADS effort began with the distinct predisposition that distributed control strategies were a definite requirement of future systems. As the study matured it became apparent that the inherent efficiencies and determinism of a centralized approach need not be forsaken. A system protocol was defined which allows the designer to implement the blend of distributed and centralized control appropriate to his needs.

APPENDIX A

MULTIBUS AVIONICS ARCHITECTURE

DESIGN STUDY

(MAADS)

INFORMAL REPORT

HIGH SPEED BUS PROTOCOLS

August 30, 1982

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The Multibus Avionics Architecture Design Study (MAADS) performed considerable analysis of high speed bus protocols with a view to identifying what is useful and desirable in future avionics systems. A number of working sessions on this topic were held with the Air Force customer personnel. These meetings revealed an intense and widespread interest in high speed bus technology.

1.1 Purpose and Scope

The intent of this informal report is to provide the Air Force complete and early documentation of the MAADS analysis effort in the area of high speed bus technology. The concentration is on the bus protocol. It is felt the bus speed will be determined by hardware technology developments and will be relatively independent of the bus protocol. This report covers only the MAADS analysis effort and the intermediate conclusion reached by the time of preliminary design review (PDR), and does not necessarily represent a final position of the program. All of the material herein as well as any follow-up analysis and conclusions will be incorporated into the interim technical report.

2.0 ASSOCIATED DOCUMENTS

A wide variety of technical documentation and articles on communications and networks were reviewed in the course of this analysis. A few of the publications are identified here. The interested reader can further explore the subject by following up on the references contained in these articles.

ETHERNET Specification; the Xerox Corporation

IEEE Proceedings of the Computer Communications Networks Conference, September 5-8, 1978

Data Bus Communications for a Distributed Processing System, E.T. Nakahara and R. Mauriello. IEEE Transactions, p. 19, CH 1402, July 1979

Multiaccess Protocols in Packet Communication Systems, F. A. Tobagi, IEEE Transactions on Communications, p. 468, COM-28, April 1980

Design Analysis of a Local Area Network, Madhav Marathe, Digital Equipment Corporation, IEEE Transactions, p. 67, CH 1586, July 1980

Packet Switching in a Multiaccess Broadcast Channel; Dynamic Control Procedures, Simon S. Lam and Leonard Kleinrock, IEEE Transactions on Communications, p. 893, COM-23, September 1975

Local Area Subnetworks: A Performance Comparison, Werner Bux, IEEE Transactions on Communications, p. 1465, COM-29, October 1981

Measured Performance of an ETHERNET Local Network, J. F. Shoch and J. A. Hupp, Xerox Palo Alto Research Center, Communications of the ACM, December 1980

Unpublished Document: Contention Network, University of Kansas

Aircraft Fiber Optic Interconnect Systems Project, Final Report (NOSC TR 576). R. D. Harder/IBM Federal Systems Division, 15 August 1980. Prepared for the Naval Ocean Systems Center, San Diego

3.0 BACKGROUND

Interest in the area of high speed bus technology is widespread and intense. Virtually every electronics/aerospace company is researching the technology of fiber optics which it is widely believed is essential to the development of a high speed bus with the performance/cost/weight characteristic suitable for flight qualified systems. The IEEE subcommittee on standards (commonly referred to as the A2K Committee) is now looking into high speed buses and it is presumed will be defining a standard to be submitted to DOD for approval and adoption for military applications.

The interest in fiber optics is more broadly based than its potential use for a high speed bus. The size and weight characteristics and the EMI immunity are significant potential payoffs in this area. No doubt, the spectacular success of the telephone company in applying fiber optic technology also has heightened the interest in this area. When the technology matures into flight worthy systems, the use of fiber optics for high speed bus applications is highly probable.

At the beginning of the MAADS effort in this area it was understood that the A2K committee was actively considering ETHERNET, a Xerox developed commercial local area network, and Free Access, a protocol developed by IBM and published as an appendix to the Naval Ocean Systems Report. Further, it was understood a protocol was in development by the General Dynamics Corporation and was to be presented to the A2K committee. As it turned out, these understandings were premature. The considerations and presentations had not already taken place and, in fact, were accomplished in the same time frame as this study effort. The MAADS effort succeeded in staying cognizant of high speed bus committee activity.

Some effort, including part of the interaction with the A2K committee, was directed at trying to identify the driving requirements for a high speed bus. It was learned that, to some extent at least, this area represents technology for technology's sake. No specific critical avionics functions were identified which demand a high speed bus for implementation. At the same time it is generally recognized that if the technology were available, applications for it would be identified in rapid order. Part of the push in this area is due to the fact that fiber optics will be pursued for other reasons anyway and thus offers the opportunity for higher speeds. Also, the quantum leap in memory speeds and processing speeds promised by the VHSIC efforts seem to be suggesting that a companion increment in bus technology is in order. In concert these technologies offer an opportunity for a new level of modularization in avionics that could revolutionize avionics systems. Finally, the clear intent of the A2K committee is to move the standardization process up front of the technology development and thus head off the problems and compromises associated with assuring backward compatibility.

4.0 REQUIREMENTS/CRITERIA

In assessing the potential impact of a high speed bus standard on the MAADS project it was judged obvious that when such a standard was available, the MAADS project or another effort further downstream in the PAVE PILLAR program would implement the standard in the avionics laboratory. The MAADS effort is, therefore, faced with the choice of waiting for a standard to be defined, planning on a retrofit when the standard becomes available, or taking a lead role in the definition of the standard. A pragmatic assessment of these options concludes that both of the last two will occur. The more substantial a contribution MAADS can make to the standard definition, the less of retrofit problem the laboratory will have to deal with in the future.

General requirements and criteria for the definition of a good standard were identified. A good standard must supply the context and mechanisms for a system designer to accomplish the design task effectively. At the same time, artificial constraints must be avoided. In a phrase, the standard must focus the design activity without limiting it.

Clearly, a high speed bus standard must support diverse applications. Purely synchronous uses may be expected as well as mainly asynchronous or "bursty" type of data flow environments. A mechanism for handling emergency messages will be required. The ability to cope with errors is an important characteristic of the standard. This applies to both normal data errors and errors occurring at critical points in the protocol. It is noted in passing that this latter case is often overlooked in the literature. It is common to find control exchange sequences or protocol handshaking which leave communications equipment in an inaccessible state if an error occurs at a critical point in the sequence.

For a high speed bus standard to be effective, it must be acceptable to a diverse group of potential users. The understandability of the standard is a key factor here. It is assumed the community of interest is basically the same group that generated/endorsed MIL-STD-1553B and therefore similarity with that standard will improve the understandability and acceptability of the high speed bus standard. A final criteria that was judged important to a high speed bus protocol was that it should be compatible with a fiber optic implementation.

5.0 ANALYSIS METHOD

The analysis of high speed bus protocols was conducted by first concentrating on the two protocols thought to be under active consideration by the A2K committee, ETHERNET and FREE ACCESS.

ETHERNET is a ten megabit local area network developed for commercial applications such as an office environment. It implements a layered protocol and features independent operation of the terminals on the network. Collision detection is implemented to account for the circumstance where two or more terminals attempt to transmit at the same time. When this occurs, the collision is detected and the terminal then immediately transmits a fixed bit pattern (jamming frame) to reinforce the collision. The intent is to assure that all terminals involved recognize that the collision has occurred. Also, the jamming pattern, while long enough to reinforce the collision, is shorter than the minimum message size permitted. Hence, the protol requires the message reception/processing software to recognize and discard message "fragments". The protocol is strictly transmitter oriented and does not permit an immediate reply or acknowledge from the receiver. Finally, it was generally understood from persons familiar with contention systems that the use of such protocols required that bus utilization had to be kept below 30% to be workable.

Once sufficient analysis had been performed to establish confidence that the protocol was understood this ETHERNET approach was rejected out of hand as wholely unsuitable for avionics applications. This early conclusion of the study was reported to the Air Force at a working meeting and received an implied endorsement.

Actually, as the study effort developed, additional attention was given to contention systems like ETHERNET. This will be more apparent in subsequent sections of this report.

The IBM Free Access proposal was also studied in depth. This protocol is targeted for a 50 megabit fiber optic bus. Manchester encoding is proposed and a command/response message format is defined. The unique feature of the protocol is a pre-message arbitration process that allocates the bus to the highest priority terminal with a pending message. With respect to 1553B, the free access allows for an increased number of terminals and longer message sizes. It maintains much of the flavor of MIL-STD-1553B, including word formats, command and status structures, and even the organization of the document. An initial judgement was made that this free access protocol could serve as a useful starting point for the development of a standard high speed bus protocol.

At about this point in the study effort it was learned that the A2K committee was only now in the process of formulating a position on ETHERNET and had not as yet received information on the free access protocol. With the MAADS effort thus relieved of its (mistaken) posture of operating in a catch-up mode, the study took on a more systematic approach.

Eight protocols were identified to be evaluated. These are:

- Collision Detection
- Time Slots
- Token Passing
- Round Robin

- Priority Polling
- Priority Overlay
- Priority Timeout
- Free Access

Subsequent sections of this report present the analysis and evaluations of each of these protocols.

These protocols represent a more or less generic set of ways of approaching the multiplex bus scheduling and use problem. They are not mutually exclusive by any means. A review of communications literature will reveal that most documented protocols represent some hybrid of the above approaches. In the analysis conducted they were treated as if each was a pure protocol in its own right. This permitted the identification of the best/worst features of each approach and an assessment of the feasibility of using the approach to define the mainline thrust of a high speed bus standard protocol.

Conversely, in the following analysis descriptions, when it is concluded that a protocol is to be rejected (or more specifically "dropped from further consideration"), this should not be taken to mean the standard should prohibit the strategy. On the contrary, the standard should accommodate as many strategies as possible to permit maximum design flexibility. Thus, "dropping" a strategy is a mechanism for narrowing the study effort and homing in on the forcing functions in defining a protocol. The final protocol recommended and the standard ultimately adopted will almost certainly represent a hybrid of several of the approaches described herein.

6.0 REVIEW OF PROTOCOLS

The following sections recap the analysis performed by MAADS on high speed bus protocols. Each protocol is presented and the major features discussed. Critical analysis judgement made are identified and a determination is made in each case as to whether the protocol is worthy of further consideration.

The analysis of the protocol was performed with a disposition to examine a number of specific attributes. Fault tolerance was considered an important factor. The protocol should be able to cope with errors, should provide an easy retry mechanism and must not have an Achilles' heel that threatens system failure if an error occurs at a critical point.

The protocol should efficiently utilize the available hardware signalling rate. The protocol should be free of unnecessary complexity, subtle control issues, and potentially expensive implementation requirements. Data integrity assurance must be provided; that is, confirmation of data transfer from the receiver should be availabe to the system designer who wants it.

The protocol must allow for synchronous systems, asynchronous system and a mixture of both approaches. It must not dictate a priority structure for message types.

The protocol must be adaptable to new technology. Implied timing relationships must be cognizant of the potentially higher speeds involved. It must be adaptable to fiber optic technology.

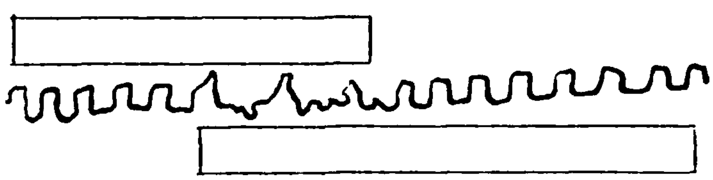
A similarity to MIL-STD-1553B is judged to be an asset since that will increase the understandability, acceptability and ultimately the speed of implementation.

Finally, the protocol must be deterministic. That is, message sequencin must be predictable and repeatable. The response to an emergency message must also be predictable. These factors are essential to support the testing an avionics system typically undergoes and to provide the system designer with control over system performance characteristics.

6.1 Collision Detection

This protocol arises when the transmitting elements of a communications net work operate autonomously. There is a probability two or more will attempt transmission at the same time, interfering (colliding) with each others' data transfer.

In its simpliest form, this protocol is implemented by letting each terminal transmit whenever it wishes and a collision may be represented pictorally as follows



The inefficiencies of this approach are obvious. The data from both transmitters is corrupted and must be repeated. As may be inferred from the sketch, even if a message is quickly repeated successfully the total time to accomplish the transmission could easily be three times the original message length. And of course there arises the concern for the possibility of repeated collision for a specific message.

The Aloha network implements this protocol and has been the subject of extensive experimentation and theoretical analysis. It is calculated that a maximum of 18% utilization of the network bandwidth may be attempted before the network stability is threatened. With higher loads, a second collision for a message has a non trivial probability. Once this does occur, the total traffic from the first collision, plus that from the second is all thrust down stream in the overall message traffic, increasing the likelyhood of additional collisions. In short, at some point the process begins cascading until all terminals in the network become involved and no successful transmissions can be performed.

Refinements of this protocol are numerous. The most obvious one is suggested by just looking at the sketch above of colliding messages. Clearly, the situation is improved if the second transmitter is smart enough to detect the presence of the first message and delay his own attempt. This approach is known as carrier sense multiple access (CSMA) and when used in conjunction with collision detection is referred to as CSMA/CD. This is the technique used in ETHERNET.

With CSMA/CD the occasion of interfering transmissions is restricted to that situation in which two terminals begin to transmit so closely together in time that neither has yet sensed the other's signal. This short time interval at the beginning of a message is known as the "collision window" and is simply due to the propagation delay of the network. The collision window is typically on the order of a microsecond in a wired network over short distances. It can range up to many milliseconds in large networks or even seconds in a satellite communication system.

The improvement obtained by using CSMA/CD is not quite as dramatic as one might expect. While the potential for interference is reduced to the short time of the collision window, a secondary effect of carrier sense is a tendancy to synchronize terminals. Since all terminals wait for a quiet network, there is an increased likelyhood they will attempt transmissions within the collision window. This thinking suggests the next variation in the protocol. A time interval, called a "mini-slot" is defined to be slightly larger than the collision window. Based on some priority scheme each terminal waits some number of mini slots following the detection of a quiet network before attempting to transmit. If a higher priority terminal exists in the network it's transmission will begin in an earlier mini-slot and be sensed by the lower priority terminal which will not interfere and simply reschedule its own transmission for a later period of time. (This process is referred to as "backoff".)

The above described protocol is referred to in the literature as slotted Aloha and is credited with allowing utilizations up to about 37% before the destablizing effects set in. Notice that the cascading phenominon of repeated collisions is still present and further in the slotted Aloha if two terminals have the same priority, i.e., attempt to use the same mini slot, a collision is absolutely guaranteed.

To circumvent these problems, a random selection of mini slots is used. If a collision is detected the terminal "backs off" a fixed time interval and reselects a mini slot surrounding the targeted transmission time. Since the terminals operate independently, two terminals which collide once will both backoff, select different mini slots (with high probability) and be collision free in their retransmissions. Should a second collision occur, the terminal doubles its backoff interval and reschedules the message. In general, if n collisions have occurred the backoff interval is multiplied by 2^n. This protocol is referred to as slotted Aloha with binary expotential backoff.

It has been shown that this approach overcomes the destablizing effect of a heavy surge of message traffic. With slotted Aloha a maximum backoff factor of 2^8 is sufficient to maintain stable operation. Additional theoretical research indicates that if the number of terminals is allowed to exceed a few thousand, it may be necessary to increase the backoff factor upper limit to 2^{10}.

Terminal populations of this magnitude are of no interest to avionics applications. Nonetheless the literature is beneficial in understanding the protocol and in particular in appreciating how it requires the utilization of the available bandwidth be suppressed to maintain stability. The reader is cautioned that the use of the item "utilization" in the literature is often misleading. Claims of up to 85% utilization will be found in some places. However, the same discussion may very well identify the average number of retransmissions as greater than three. The effective utilization is still down in the 25% to 30% range. It is well to remember also these figures include all the source/distination addressing and other header information required in the protocol. The utilization as seen from a users' data throughput viewpoint, is even less.

Other indications are that even the above figures are optimistic. Design documentation on a CSMA/CD network which was reviewed indicate a design goal of utilization below 20%. Estimated data flow requirements indicate initial loading of about 5%. The article on performance measurements on an ETHERNET reveal even a lower set of numbers. The average load in a 24 hour day is identified as 0.8%. Averaging over six minute intervals, this same data shows a peak loading of 7.9%. The more instantaneous loading conditions are displayed by averaging over one second intervals. This shows a worst case of 32.4% with an overall average of 2.7%.

Translating this last number into effective data throughput, one arrives at a performance figure of about 14 Kwords/sec, a figure that is well within the capacity of a single MIL-STD-1553B multiplex bus. The usefulness of a CSMA/CD protocol for avionics is highly doubtful.

The important factor to recognize in that the CSMA/CD protocol is directed at a system of highly autonomous user terminals; definitely not the case in avionics. The not too surprising conclusion of this analysis is that given a significantly different problem, one should expect a different solution.

Another characteristic of collision detection protocols is that message sequences are necessarily uncontrollable, hence unrepeatable and therefore very difficult to test. This is not in keeping with the comprehensive testing normally performed for avionics systems.

A final consideration relating to collision detection protocols is that the actual collision detection process itself may not be feasible. In the slotted Aloha examples discussed above, it was indicated two transmitters would detect each others' transmission and both backoff. But if in fact the signal from the first transmitter is just reaching the second terminal when it begins to transmit, this terminal may quickly detect the collision and abort his own. The result could be a very short period of interference from the second terminal. This brief signal is attenuated as it returns to the first terminal and there is no clear guarantee that it remains detectable. It is interesting that only ETHERNET anticipates this problem and institutes the jamming pulse train to assure collision detection.

Part of the ETHERNET literature points out another interesting case. Often the carrier sense function is implemented by detecting the phase shift in the waveform. But if multiple transmitter attempt to use the bus simultaneously, it may result in current saturation, holding at a constant level. A saturated bus then looks like an idle bus, effectively inviting other terminals to join the traffic jam.

Collision detection in a fiber optics network is possibly an even more difficult problem. The dynamic range of fiber optic receivers is already an area of concern. The "listen-while-talk" requirement of collision detection adds the need to be able to handle the signal from the nearby (it's own) transmitter and yet to be responsive to the distant signal from another unit. It is also conjectured in some of the literature that fiber optic receivers that are required to be on while the (necessarily close) transmitters are functioning will have very short lifetimes, significantly impacting maintenance and life cycle costs. (Note, this is the phenominon that leads to the suggestions of transmissive star couplers, a multi-fiber approach that logically appears to be a bus structure.) There exists, therefore, some genuine doubt that a collision detection protocol can readily be transitioned to fiber optic technology.

To summarize then, the analysis of collision detection protocols leads to the conclusions that they require utilization be kept low in order to work well; they will cause significant testing problems due to undetermined, unrepeatable message sequences; and they may not be easily upgraded to new technologies. The collision detection protocols are therefore dropped from further consideration.

6.2 Time Slots

A time slot protocol is one in which the use of the transmission medium is pre-allocated. Each of the terminals in the system knows the time it is permitted to transmit and it waits for the time, takes control to transmit (or receive if the protocol permits this) does its own thing and then relinquishes control at the end of its time slot. This may be pictorially represented as follows:

1	2	6	3	5	2	4

This protocol approach is also known as time division multiple access (TDMA), or sometimes as "pure TDMA" since the time division is the only basis of control transfer identified in the original statement.

This protocol has a strongly synchronous flavor to it. With a purely synchronous application, all message sequences can be predefined in some optimum fashion. Once a system wide time base is established the terminals can take their turns managing the data flow assigned to them and the control transfer from one terminal to the next can be as rapid as the clock resolution permits. In principle, this protocol can approach 100% bus utilization. Time slotting is highly fault tolerant in the sense that if a potential controller fails, the system continues to operate with the other terminals performing data transfers during their assigned slots. In effect the slot for the failed terminal just goes blank.

The time slot protocol is less fault tolerant when individual message errors are considered. The baseline definition makes no allowance for message retry. If slots are fully assigned and tightly packed (i.e., designed for 100% utilization) the protocol must explicitly prohibit message retry; message errors are basically ignored.

This concern for message retry generates a first variation on the time slot protocol. The slots are oversized relative to the message traffic required in order to reserve a certain fraction of time for message retries. The penalty of course is reduced efficiency. The system designer can elect to reserve enough time to allow all messages to be retried once. He does so however only by driving the efficiency down to a 50% maximum.

In between these two extremes (100% use and 50% use) the system designer may select whatever value is deemed optimum for his system. But now a new concern arises. Once message retries are permitted, but time is not reserved sufficient to retry all, there then exists the possibility of a time slot overrun. To manage this problem, logic (probably software) must be added to make determinations about extending the time slot or truncating message retries in order to stay inside the assign time.

Extending the time slot requires now that the next potential controller (and therefore all controllers) do something like monitor bus traffic prior to initiating messages. On the other hand, truncating retries in order to maintain the slots leaves the retry strategy less reliable. In short, there is a basic message retry versus efficiency tradeoff to be made and system complexity begins to rise as one moves away from the pure TDMA.

Time slots do not easily accommodate asynchronous message. First, there is the question of allowing time for them. Like message retries, some reserve allocation must be made. And again, either this allocation is very generous (with considerable efficiency impacts) or else the time slot overrun must be dealt with, introducing attendant complications.

Given the above, the response time provided the asynchronous messages is still not very good; that is the emergency message is not well handled. Basically the terminal in which an asynchronous message arises must wait for its next available time slot in order to transmit the message. This problem can be attacked by giving the source terminal frequent short time slots. This, however, is just another way of allocating reserved slot time and it has the same overall system effect.

Another variations on the time slot approach consist of dynamically assigning the time slots. For example the last terminal in a major frame can poll other system elements and plan the next set of slot assignments and broadcast them to other terminals. This approach is much more responsive to a dynamic environment and gives improved handling of emergency message. There is more overhead involved and there are some unpleasant fault tolerance implication. The dynamic slot assignment process becomes a single point of failure and the message communicating the slot assignment becomes a critical message; that is a message that must succeed in order for the system to function correctly.

In summary, the strongly synchronous, very clearly defined time slot approach offers outstanding performance for a highly synchronous system. As deviations from that are accommodated by the protocol, efficiency impacts are accumulated and control complications are introduced fairly rapidly. The time slot protocol is therefore judged to be unsuitable for a general baseline approach and is dropped from further consideration.

6.3 Token Passing

This protocol consists of a terminal performing bus control to accomplish its data flow requirements and at the completion of those operations, sending a special message that transfers bus control to another terminal in the system. This special message contains a data word called a token identifying what terminal is to take control of the bus. The offering terminal at the completion of his operations simply takes the token message as he received it, adds one to the token value and sends out the message.

This elegantly simple control transfer mechanism accomplishes a number of things more or less automatically. First, recognize that when the last terminal to administer control completes its operations a token message is formulated and sent out with a non-existent token number. No terminal takes control, so there is a brief lapse in the data flow. That terminal currently assigned token zero is charged with the responsibility of timing out on this lack of bus activity and starting its own period of bus control. As noted above when those messages are completed, control is then passed to token 1. The protocol automatically restarts itself with token zero regardless of the number of tokens currently active in the system.

A terminal coming on line to an already active system simply has to monitor the system for a few cycles to see what token message ends each cycle. When no terminal responds to a specific token message, the terminal trying to enter the network appropriates that token number for his own. On the next cycle (or as many as needed to establish the correct token number with some confidence) the terminal responds positively to the token message by initiating his own set of messages and bus control functions. Since this is done promptly, the token zero terminal does not restart the cycle until the new terminal has completed operations, passed on the token, and no other terminal responds to that.

With these defined mechanisms, consider now what happens when a terminal suddenly fails. If part way through a cycle, the token is offered to a terminal that has failed, the token is in effect, "dropped". No terminal takes control and bus activity ceases. When this occurs, the terminal with token zero functions as usual, detecting the lack of his activity and restarting its own period of bus control. The failure of a terminal with a given token causes all higher numbered tokens to be skipped. Logic in these terminals is required to recognize and respond to this situation.

Recognition of this situation is a matter of the terminal timing out on the interval since it last received control. When more than two full cycle times have passed without the terminal receiving the token offer, it decides something has failed in the network. The response the terminal makes at this point is to decrement its token number by one. On the next cycle the terminal "picks up" the "dropped" token and normal operation of this and higher numbered terminals (which have performed the same process and decremented their own tokens) may now resume. The network response to the failed terminal situation is to run a few abreviated cycles which effectively confirm the failure and then to close the gap and resume normal operations without the failed unit. When and if the unit recovers it may attach itself at the end of the loop as previously described.

It is to be noted that the above described mechanism works even for the case of a failure of the token zero terminal. After a period of time, the token one terminal discovers it is not being serviced, decrements its token to zero and assumes the function of starting each cycle. This migration of token number in response to failures implies that all terminals must have the capabilities defined above for the token zero terminal.

The token passing protocol is designed to be highly fault tolerant of controller failures and clearly has achieved that objective.

The approach does not, however, easily satisfy the requirements of a synchronous system. The failure of a terminal in the loop causes the data from that and all higher numbered tokens to simply stop for a while, and then resume operation with a portion of the data flow missing. Subsequent recovery of the terminal may reinstate the missing data but at a different place in the overall cycle. The synchronous system practice of scheduling data flow and task execution with a fixed time relationship would not be reliable.

To try to maintain such a relationship it would be necessary to handle it somewhat like asynchronous tasks. That is, the data arrival could be treated as an event which in turn could be used as a condition for task execution. To accomplish this, software inspection of the data received might be necessary.

Possibly with a careful system design, these problems could be avoided by structuring a strictly receiver oriented message flow. But even then the implication remains that task processing can be reassigned on the time line. This raises a system level issue of whether the designed distribution of processing loads can be maintained.

Neither does the protocol offer a good environment for managing asynchronous operations. Basically, regardless of when the requirement for an asynchronous message may arise, the terminal cannot transmit the message until the token is passed to it. The response time provided asynchronous messages will, in general, average half the total cycle time of the system. But since a terminal can be skipped due to problems with another terminal, not even this time can be guaranteed. A true emergency message, that is, an asynchronous message with a very short response time requirement cannot be handled by the protocol. Some add-on such as frequent polling of the source of such messages might be able to achieve the necessary response. Relatively large overhead impacts may be expected in such an approach.

Another area of concern is the impact of errors on the token passing process and vice versa. It is to be noted that the time out executed by the token zero terminal should be kept small in the interest of efficiency. This time out interval, whatever it is defined to be also defines, necessarily, the maximum time any bus controller may pause during its operations. Should a controller, due to some special situation such as error analysis take too long before its next bus operation there is the possibility that the token zero terminal will interpret this as the end of a cycle and start the next cycle.

When the pausing terminal attempts to resume operations it will now collide with the traffic from the token zero terminal. The normal result of colliding terminals is that both believe they have failed. If this occurs the entire system stops until the other controllers recognize the problem and adjust their tokens. Even at this point the difficulty hasn't been resolved. When the two failed terminals attempt to rejoin the network they will likely collide again. Another possibility, depending on the relative timing in the various terminals, is that one of these recovering terminals could mistake a gap in the network for the end of the cycle. In this case it would appropriate a token already in use and when it attempted to reenter operation it would precipitate the apparent failure of yet a third terminal.

Another potential outcome of the original pair of colliding terminal is that they succeed in establishing apparently normal operations but on separate redundant buses. This eventuality would have less immediate failure impacts but would lead to protracted erratic system operation with the problems occurring at the individual message level.

These kinds of considerations would probably lead to stretching out the defined interval for the token zero time out and some set of rules for sampling bus activity prior to starting a new cycle. These factors along with some estimates of overall system load would then need to be input to the process of defining the time interval that each terminal would use in deciding when to decrement its token. This would have to be sized for the maximum case and more than likely this time interval would also have to be exagerated in the interest of caution.

A more pragmatic approach might be to rethink the token passing handshake with a view to making it more ironclad and of detecting a dropped token more quickly. Perhaps for example the message should be "terminal X passing the token to terminal Y with terminal Z selected to validate the handover". A procedure could be developed for terminal X and terminal Z to cooperatively determine when terminal Y had failed. This information could then be communicated to the rest of the system. In general, the more widely distributed the total system state information is, the more reliable the overall operation. Voting and cross checking can be initiated to isolate failures with certitude.

At this point in the analysis it even becomes doubtful that the nebulus token should be maintained. A direct use of terminal addresses, total system state information and cooperative validation procedures should produce a more manageable and predictable operation.

In summary then a token passing protocol while apparently having good fault tolerance characteristics, does not support synchronous operations, does not provide a good environment for asynchronous operations, cannot handle emergency messages, and in the final analysis is plagued by subtle control procedure issue associated with duplicated tokens and propagated errors. This protocol is dropped from further consideration.

6.4 Round Robin; Dynamic Bus Control

This protocol uses the 1553B type of mode code operation, dynamic bus control to pass control from one unit to another. Each controller admisters the data flow as required for its own resident applications and at the end of this executes a bus list of mode code operations offering bus control to each of the other controllers in the system. The list execution terminates when another controller accepts bus control. The dynamic bus control mode code operation as defined in 1553B requires a positive response form the terminal to which control is offered. This response consists of a status word with the specific bit, "dynamic bus control acceptance bit" set to one. The bit being set to zero indicates a decline of the bus control offer. If all the other potential controllers decline the bus offer the current controller keeps control. Depending on the system design it may then proceed to other useful message operations or repeat the mode code list offering control to the other units.

This round robin bus control protocol may be viewed as a variation on the token passing protocol. The principle difference is the positive acknowledge provided by the status word response. Also the mode code list structure provides an automatic sequencing to the next potential controller when an offer is declined. This minimizes the overhead of the control hand-over. It can even be performed by an intelligent bus control interface (BCI) so that the host processor need not be involved.

The one problem case in the control handover of the round robin protocol is the situation in which no status word response is received. This is treated as an exception condition by the BCI so the automatic sequencing is terminated. But the situation is ambiguous. It is not clear what has occurred. Interference on the bus may have garbled either the command or the status response. If the command was garbled, the other terminal does not recognize that bus control was offered, or even that a command was directed to it. It therefore, has made no response and is not initiating bus control. If the status was garbled, the terminal may have correctly decoded the command, accepted bus control and made (from its point of view) a proper status response. The terminal may be completely unaware of any problem and could be on the verge of issuing its first bus command. The offering controller cannot, therefore, safely even inquire as to the status of the last operation for fear of colliding with the operations of the new controller.

The obvious solution to this difficulty is for the offering controller to cease operation on the detection of the no status exception condition and to enter a mode of monitoring the bus for activity. A fixed time out interval must be defined during which bus activity must begin. The control handover is judged a success if activity is detected. If no activity is detected the controller may resume the process of offering bus control to other units. To prevent repetition of this occurrence the controller might first take the time to access the status of the terminal that did not respond previously. One possibility is that the unit has failed and is not communicating at all. In this case further attempts to offer bus control to it would be fruitless.

The support of synchronous message operation is easily envisioned with a round robin protocol. Each minor cycle, each controller could set up a synchronous instruction list (SIL) as the first message traffic to be performed. Each SIL would then be followed by a mode code list of dynamic bus control commands. The first controller could perform its own SIL then pass controller to the second unit to do its SIL and so forth. When control is finally returned to the first unit asynchronous operation may be accomplished. This list also may be followed by a list of dynamic bus control commands and control can be passed around the system again to accomplish asynchronous message operations. When control returns to the first unit for the third time, status polling may be initiated or whatever low level background message traffic is defined for the system can be performed. If desired, the units can continue to rotate control around the system until it is time to start a new minor cycle.

As indicated above, asynchronous messages may be accommodated by the rount robin protocol. In the above discussion it is assumed asynchronous messages are handled as a lower priority operation than synchronous. Reversing the priorities is easily accomplished as well. That is simply a matter of the sequencing of the bus list. It is also to be noted the above example implies a transmitter oriented asynchronous message protocol. That is, each controller transmits those asynchronous messages arising from its own host operation. Asynchronous messages originating in other than potential controller pose a slightly different problem.

Such messages may be accommodated by requiring the originating terminal to employ the service request (SR) mechanism as defined in MIL-STD-1553B. The controller currently in charge of the bus operations must recognize this exception condition and respond to it. Exactly what this response is, represents a key system design decision. If the vector mechanism is employed to identify the specific asynchronous message, then the question arises as to where are the tables resident which permit the vector to be translated to the bus operation to be performed. If the tables are distributed, that is, if each controller only knows about a portion of the asynchronous messages, then the protocol creates the possibility that the controller responds to the SR exception and performs the vector word operation only to find out it cannot perform the asynchronous message operation. In a DAIS like implementation this has the further difficulty of the vector operation suppressing the SR bit. The request would thus be hidden from other controllers.

If the vector mechanism is employed therefore, it seems wisest to require the requisite tables be maintained in all controllers. Even this is not a very satisfactory solution. Wieh the bus instruction formats as defined in the MAADS I/O specification, the tables cannot contain the actual bus instructions and still be identical copies in the separate controllers. The table entries must be encoded values to be translated to bus instructions or else bus instruction entries must be reformatted based on which controller holds the table.

A much cleaner implementation of this kind of asynchronous message operation is possible if the unit originating the asynchronous request is required to have bus transmission control capability. Then, when the SR bit exception is recognized the response could simply be to issue a dynamic bus control mode command to the requesting unit. This unit can then perform the desired operation and return control.

With this protocol, it does not matter which controller is in charge of the bus when the service request arises. The vector mechanism is not needed and the tables to translate it no longer exist.

The requirement that the requestor be capable of transmission control is not judged to be a serious impact. No new hardware is needed. Some small memory requirement is added plus the firmware to access it. The sequencing of words out of the terminal is actually very similar to what occures in a standard terminal to terminal operation. A command word simply replaces the transmitter's status word in the sequence.

The one area in this protocol enhancement that needs additional analysis is the error management. The question is; should the transmitting terminal be responsible for determining the correctness of the operation and managing retries. The alternative of letting the original controller do the error management raises the prospect of doing status word reception/analysis for an operation that the unit is not controlling. None of the obvious choices seem very appealing. Additional thought needs to be given to the full implications of the various options. In particular automatic retires by the transmitting terminal must be coordinated with the procedure adopted for the original controller to resume bus control operations.

Though issues like the above must still be worked, there is no doubt the round robin protocol can accommodate asynchronous message operations. It is to be noticed however that no type of priority is supported. Even when a controller is polling terminals, looking for service requests, message operations are performed as the need is encountered, not in some priority controlled way. A priority scheme could be imposed on the protocol but only by insisting that all potential sources be polled before any bus operations are initiated. Predefined message priorities could be implemented or the vector word could be adapted to a priority code.

Emergency messages are essentially high priority asynchronous messages and require a polling operation. If the sources of emergency messages are equipped with a bus control capability, one version of the polling operation could be to include these devices in the dynamic bus control mode code list. This is an area requiring the careful attention of the system designer. It is axiomatic that a very quick response time for emergency messages can be accomplished only with the expenditure of considerable system overhead.

An interesting aspect of the round robin protocol is that with only the implementation of the dynamic bus control mode code (which Notice 1 prohibited in Air Force systems) the protocol could be demonstrated in the AVSAIL facility on a 1553B type of bus. This of course would not be at the high speeds of a wideband bus, but the control strategy could be exhibited.

In summary, then, the round robin bus control protocol using the dynamic bus control mode code provides an effective means of distributing bus control with a positive acknowledgement of the handover process. It accommodates synchronous and asynchronous system approaches or a hybrid of the two. Emergency messages can be handled, although very rapid response times can be accomplished only with a large overhead or special approaches such as requiring the sources of such messages to have a bus control capability. The analysis effort concluded the round robin protocol should continue to receive consideration and should be carried forward into the throughput analysis phase.

6.5 Priority Polling; Dynamic Bus Control

This bus control protocol consists of the current controller completing its operation(s) and then passing control to another controller via a dynamic bus control mode command. Unlike the round robin approach, control is not simply passed to the next controller in sequence. Instead, special actions are taken to determine which potential controller has the highest priority message pending.

The special actions consist of issuing a transmit status mode command to each of the potential controllers. The status response is defined to include a priority field which contains the value of the highest priority message pending in that controller. After the polling is completed, a simple algorithm determines the unit with the highest priority message and issues a dynamic bus control mode command to that device. This controller then performs the high priority message operation(s) and then initiates a new polling sequence to determine where to pass control next.

It is a matter of system design whether a single message is performed before polling or a group of messages is completed. Clearly, doing a group of messages produces higher throughput characteristics. Polling before every message yields better response to emergency message requirements.

After the control passing mechanism is defined, this protocol can be a command, response approach like MIL-STD-1553B. Virtually all the analysis discussed under the round robin protocol is equally applicable to this priority polling protocol.

Of special note, this priority polling, like round robin is demonstratable on current day equipment. All that needs to be defined is how to transmit a priority. The vector word could be used for this purpose in which case transmit vector mode codes would be used rather than transmit status mode codes.

Looking to future technology, it also is to be noted that a more intelligent BCI such as that defined in the MAADS I/O specification would be capable of conducting the priority polling operation on its own automatically.

Priority polling algorithms for the purpose of allocating a resource of some kind are not by any means new. They have been studied at some length.

One of the principle characteristics of priority polling is that it yields reasonable performance with small numbers of units to be polled, but with large numbers of potential controllers, the overhead escalates very rapidly.

Nonetheless, as mentioned above, most all of the analysis of the round robin protocol applies equally well here and the conclusion is the same. The priority polling protocol is carried forward into the throughput analysis phase for additional consideration.

6.6 Priority Overlay

As with several of the previous approaches examined, this is not really a complete protocol, but rather a mechanism for establishing what unit is controller of the bus. After control is established, the protocol administered can be any of a number of techniques. For the purposes of the MAADS analysis a 1553B like command, response approach was assumed. Thus the analysis effort was permitted to concentrate on the unique aspects of the protocol.

The priority overlay is a pre-message arbitration of bus control. It is in some documentation referred to as "transparent contention resolution/collision avoidance". A version of the priority overlay protocol appears in the interim report from the fault tolerance computer network study.

The basic concept is that all units contending for the bus transmit a priority code in the form of a string of binary bits. The electrical characteristics or the bit representation technique is such that if one unit transmits a one bit and another unit transmits a zero bit, the result on the bus will be a one bit. In effect, what appears on the bus is the logical OR of all the units' transmissions.

Each unit also reads what is on the bus at the same time as it is transmitting ("listen while talk"). As long as the unit is able to read the same value it is trying to transmit it remains in contention and preceeds to the next bit in its priority code. If the unit attempts to transmit a zero, but reads a value of one on the bus it drops out of contention and ceases the transmission of its priority code.

By starting with the most significant bit position first and working through the bits in descending order it is evident that the unit with the highest numerical value initially will be the only one to survive the process and hence win the contention. Note that it is important to the process that each unit drop out (cease transmission) immediately upon detecting a miscompare. That is, it must not transmit another single bit.

Also, it is implicit in the process that no two units may start with the same code value. Thus, the terminal number may be used as the priority code or each and every message in the system could be assigned a unique priority. A hybrid approach of concatenating a message priority with the terminal number to form the priority code value is also possible.

The final key specification for this protocol is how the priority overlay process gets started. The common choices are a special message to signal all terminals to begin the contention process, and the initiation of the process a fixed time after the end of a prior message. The special message is the preferred approach since it leaves the system designer the option of having a terminal transmit multiple messages once it acquires control of the bus. Also it clarifies the proper handling of the situation in which no terminal has a message to send. The terminal that sends the original message, if it has no message of its own to contend with, creates an artifical code of all zeroes. If it wins the contention, it simply rebroadcasts the special "begin contention" message.

The analysis of the priority overlay protocol is fairly quick and straightforward. The underlying assumption that individual bits from separate terminals can be overlayed is no longer valid at the bus speeds and bus lengths of interest. For a ten megabit bus an individual bit time is 100 nanoseconds. For a one hundred meter bus the round trip propagation delay is approximately one microsecond.

Due to the propagation delay of the special message each terminal in the system begins transmitting at slightly different times. The signals from these terminals suffer the same propagation delay and thus arrive at other terminals at different times. Thus, the bits do not overlay and OR together on the bus as intended. What actually appears is a train of signals that interfere in some indeterminate manner and, in principle, appear differently at every terminal on the bus.

The conclusion, therefore, is that the priority overlay simply will not work.

The obvious adjustments required to this protocol are twofold. First, the spacing of the bit transmissions from a terminal must be based on the propagation delay time rather than the normal bit time. Second, the detection of other terminals' transmission cannot depend on a hardware OR function and must be extended in time to account for the propagation delay. The "listen while talk" must be replaced by a "talk, then listen for a while" function.

The free access protocol discussed later makes a movement in both their directions in the way it approaches contention resolution. Hence, there is no purpose in giving special attention to priority overlay as a separate protocol. Priority overlay is dropped from further consideration.

6.7 Priority Time Out

This bus access technique concerns itself with the problem of independent terminals accessing the bus in such a way as to avoid collision altogether. Each terminal monitors the bus operation, detects the end of the current message and starts a timer. The terminal continues to monitor the bus and any detected activity resets the timer. If the timer runs out and no bus activity has occurred, the terminal takes control of the bus and transmits (or receives) its message.

The key to the protocol is that the timer interval is based on a priority value. Again, there is the requirement for absolutely unique priority codes whether based on terminal address, message priority or a combination thereof.

A variation on this is to allow the priority codes to overlap and then do a collision detection type of protocol. (Note the basic priority time out already has the carrier sense feature built in.) The strategy here is to use the priority time out to reduce the chance of collision and then use the collision detection to resolve the conflicts that do occur. This is in fact the technique discussed before under the title of "slotted Aloha".

In the pure priority time out, the strategy is to avoid the collisions altogether. To achieve this the requirement is that two side by side priority codes result in timer values with a difference sufficiently large so that the higher priority can time out, initiate a message operation and have the second terminal detect the resultant bus operation before its timer expires. When it is recognized that the timers may be initiated at different instants due to the propagation delay, it is clear that the units of time must at least be equal to the round trip propagation delay.

For a typical avionics bus on the order of 100 meters or less this round trip time is about one microsecond. Thus, with a priority time out protocol the highest priority message could be sent in say, three microseconds after the bus is free. The next level could be sent at four microseconds, the third level at five microseconds and so forth.

Another element of the protocol is the determination that the bus is free. As discussed under other sections, the best approach is an explicit message to that effect. It would be possible to simply operate on bus activity, but there is then a concern for gap times built into the defined message formats. For example, in 1553B a terminal is allowed to take up to twelve microseconds to reply with its status word. If this is the case and only bus activity is being sensed, the base interval time (the arbitrarily chosen three microseconds above) must be increased to at least 13 microseconds.

The explicit "bus free" message also is used to cover the case when no terminal chooses to transmit a message. The terminal originally transmitting the message, if it has no message to send itself, load its timer with a value greater than that required for the lowest priority message and then if the timer expires, the "bus free" message is broadcast again.

It is important to notice the influence of the lowest priority message on the overall system. In the context of the prior example, suppose the bus free message has just been sent and a specific terminal did not have any message to offer. Now, just after the time out process has begun, the terminal developes a requirement for a high priority message. This new high priority message must wait for the next bus free message to enter the process. This bus free message repeat is paced by the requirements of the lowest priority message whether or not such a message is currently pending in the system.

The concern with the priority time out protocol is that the time outs required for lower priority messages can become quite long when a large number of messages are involved. In order for such messages to have an opportunity to be transmitted there must be regular occurrences of dead time on the bus equal to or longer than this period. This can be accomplished only with severe impacts to overall system utilization.

It is important to recognize that the priority time outs to be utilized derive from the basic signal propagation time and are independent of bus speed. As the bus speed is increased message transmission times are decreased. The ratio of delay time to transmission time then increases for higher bus speed. That is the priority time out protocol's ability to effectively utilize the bus goes down with increasing bus rates.

The priority time out protocol is clearly designed for asynchronous operations and should give good average response times to emergency messages. The guaranteed response time, however, is limited by the longest message at the lowest priority. As we have seen, the time out delays needed depend on the total number of messages in the system. If, as expected, avionics systems generate requirements for hundreds of unique messages, the delays involved are well beyond the limits of acceptability.

The priority time out protocol is thus not considered a good candidate for future systems and is dropped from further consideration.

6.8 Free Access

This protocol is the one thought to be under consideration by the A2K committee during the early phase of the MAADS effort. It was therefore, given early and detailed attention and where it appeared necessary, certain modifications were identified.

A review/critique of the free access protocol was separately documented in TRW memo H122-S52-14, 1 March, 1982. That information is not reproduced here except for a summary of the highlights of the specification which are as follows;

- Basic Data Rate: 50 MHz
- Manchester Encoding
- Word size; 20 bit times; therefore, 0.4μs per word
- Single Word sync; 2 bit times
- Contention Resolution prior to message transfer; based on terminal priority
- Message Transfer; Command Response Protocol
- Control Words; 4 kinds:

  Address
  Command
  Status
  Priority (Poll)

- Commands; 4 Types:

  Immediate
  Control
  Short Data Block
  Long Data Block

- Data Blocks; "short" up to 256 words
  "long" TBD

- Terminals/Subaddress;
  up to 64 terminals/256 subaddresses per terminal

- Transmission Technology; both electrical and optical are specified.

As may be inferred from the above, the specification has the flavor of extensions and modifications to 1553B as applied to a much higher speed bus. Even the terminology and document organization are very much "1553B-like". It was concluded that once the access contention process was understood, anyone familiar with 1553B would assimilate the protocol easily.

While all aspects of the protocol are important, most of the details can be modified without affecting the overall performance characteristics. The following discussions therefore concentrate on the contention resolution process.

The terminal completing the current message is responsible for administering the contention process. As always, one or more other terminals in the system are responsible for monitoring the bus for excessive dead time and are prepared to initiate the contention process if needed.

The contention process is performed by a series of polling (or "priority") commands. This is a control word with a specific command code and a priority mask field. Within each command issued the priority mask field has one and only one bit set. The first poll command has the high order mask bit set. Each terminal receiving the command compares the mask field to its own priority code. If the high order bit of its code is set, the terminal makes a special response consisting of a unique sync pattern which is three bit-times in length. The terminal then waits for the receipt of the next poll command.

If, when the comparison with the terminals priority code is made, it is found the high order bit is not set, the terminal makes no response on the bus. The terminal then monitors the bus for a short while to determine if any other terminals are making a response to the poll. If no response is detected, the terminal simply waits for the receipt of the next poll command. If any response is detected on the bus, the terminal "drops out" of the contention process and will not respond to subsequent polling commands.

The terminal administering the process then issues the second poll command, this time with the second-most high order bit set. Only those terminals that made a positive response to the first poll are eligible to response to this poll (unless nobody responded to the first poll, in which case all are still in contention.) The terminals proceed as before, responding and remaining in contention if they have the indicated (in this case, second) bit set. If they do not have the second bit set, they do not respond, but do listen to the bus, and as before, drop out of contention if they detect any other terminal respond.

The third poll is then issued with only the third high order bit set and so the process continues until polls for each of the six bit positions have been issued. At the end of the process one and only one terminal will remain in contention. This terminal wins the contention process, takes control of the bus and performs its message operation. At the end of its operation this terminal then initiates the next polling cycle.

As is evident this process is conceptually similar to the priority overlay technique. Bit by bit, each terminal is checking to see if it has the highest priority in the system. The free access protocol is an improvement in that an allowances is made for the propagation delay, and a precise overlay of responses is not needed. Also since a terminal that responds positively to a poll is not required to listen to the bus, it may be that the dynamic range problem can be avoided if the receiver can be switched off quickly enough.

The allowance made for propagation delay is actually not sufficient. The "round trip" is not correctly taken into account. Neither is there a clear appreciation of the relative times involved. The specification makes mention that the sync response may appear "elongated" due to propagation delay. However, the time window for listening after not responding is likely on the order of one microsecond whereas the three bit-time sync response on a fifty megabit bus will be sixty nanoseconds. The "elongation" is more likely to be a train of waveforms that, as in the priority overlay, may appear (interfere) differently at every terminal. Theoretically, in an electrical transmission they could even cancel, but that is viewed as a statistical impossibility.

The more practical concern is the design of the receivers. With this free access protocol, the receivers are required to meet all of the following simultaneously;

- not be damaged by a nearby transmitter

- by sensitive to a sixty nanosecond waveform from a distant transmitter in spite of possibly destructive interference

- not be confused by potential reflections of the original polling command.

Thus, with the free access protocol, there are some technology hurdles yet to be overcome and it is not at all clear the approach will work at the specified speed. Nonetheless the analysis was persued.

The initial evaluation of the free access protocol concluded that a priority system based on terminal number was not going to be satisfactory. Therefore, a three bit message priority was defined to precede the terminal number for the purpose of the contention resolution process. In addition, a special poll command was defined. The terminal administering the contention would place its own highest priority requirement in this special command and in effect broadcast the question "Does anybody have a higher priority"? If no response to this inquiry is received the terminal keeps control and performs the high priority message.

At one point in the analysis it was thought this type of poll could be used for the entire contention process. The intent was to do a binary search for the highest pending priority. It was also thought this would give the system designer the flexibility, by selecting the search starting point, to skew the system performance in favor of fast response (by starting with a high priority) or in favor of average throughput (by starting with a low priority) in accord with what was deemed in the best interest of a particular system.

Subsequent analysis of the "Anybody Higher?" poll with a binary search on priority showed that this approach suffers from a problem similar to the priority time out approach. For a moderate number of priorities there is always some case for which the contention resolution becomes extremely long.

This approach was therefore discarded but the leading "Anybody higher?" poll was kept. Part of the rationale for this is that it serves to alert all terminals that a contention process is to start. Recall that the original definition has terminals drop out of contention, disabling themselves from responding to subsequent polls. The protocol does not include a re-enable function. The "Anybody Higher?" poll can serve this purpose.

A concern developed that extending the length of the priority field and adding an extra front end poll operation would add to the contention resolution time and unfairly penalize the free access protocol. The response to this was to reduce the terminal address field. It was thought the original free access (implied) requirement that all terminals had to be controllers was not realistic. A better guess is three to seven controllers in future avionics systems. A three bit terminal address field was adopted.

In summary then, the initial evaluation of the free access protocol was that it was a reasonable starting point. Subsequent analysis identified some technical difficulties but no fatal flaws. The fifty megabit speed specification was not accepted as an integral part of the protocol. Some modifications of the protocol were defined. The free access protocol as modified, was carried forward for additional consideration.

7.0 ATTRIBUTE ANALYSIS

For presentation purposes, an attribute analysis chart was developed. This identified nine desirable attributes of high speed bus protocols and then rated each of the reviewed approaches. This represented a consolidation and cross check on the preceeding analysis, and offered the opportunity to give the analysis a more quantitative flavor. An ordered ranking of the protocols was accomplished for each attribute. A numerical value in the range of one (very poor) to ten (excellent) was assigned according to how well the protocol exhibits the specified attribute.

After review with the Air Force, an entry for MIL-STD-1553B was added as a reference point and the exercise repeated. The final result is summarized in Table 1 and discussed in the following paragraphs on an attribute by attribute basis.

Fault Tolerance. This represents the protocols' ability to cope with errors, including those occurring during a control handover sequence and especially the protocols' ability to function in spite of a bus controller failure.

In this category, token passing is placed at the bottom of the list due to the perceived inclination of this approach to permit errors to propagate to multiple unit failure. Time slots are rated next lowest due to the inability to accommodate retries without substantial efficiency impacts. Free access and priority overlay are treated as similar both being vulnerable to simple errors occurring in the arbitration process. The MIL-STD-1553B protocol is given a moderate rating due to its implied reliance on a centralized controller, while the remaining approaches are ranked higher due to the explicit definition of multiple controllers. Round Robin and priority polling are put at the top of the list because of the positive acknowledgement contained in the control handover sequence.

Efficiency. This factor measures the protocols' ability to make use of the available bandwidth for user data flow. Here, 1553B goes to the top of the list since it has no control handover overhead. Round robin ranks second because of its direct approach with token passing just behind. A group of the protocols are rated as equivalent in this area with time slots and priority time out downgraded slightly. Collision detection is put at the bottom of the list for efficiency considerations.

Simplicity. The concern here is really complexity, but it is made a positive factor by ranking on the basis of the absence of complexity. Time slots is clearly the simplest protocol. Round robin, 1553B, and priority polling are judged to be about equivalent. Priority overlay and free access are ranked next with token passing just behind and priority time out and collision detection placed at the bottom as the most complex of the protocols.

Data Integrity. This is an estimation of how well the protocol assures correct data transfers as opposed to permiting undetected errors. Most of the protocols are judged fairly good in this respect. The middle three are given a slight edge because of the required command, response sequence. Time slots is downgraded slightly because of the retry problem and collision detection trails the list due to the real possibility of data corruption.

TABLE 1. PROTOCOL ATTRIBUTE SUMMARY

PROTOCOL	FAULT TOLERANCE	EFFICIENCY	SIMPLICITY	DATA INTEGRITY	SUPPORT SYNCHRONOUS	SUPPORT ASYNCHRONOUS	ADAPTABLE TO NEW TECHNOLOGY	SIMILARITY TO 1553B	DETERMINISTIC
	ATTRIBUTES								
COLLISION DETECTION	7	3	3	4	2	6	2	2	2
TIME SLOTS	4	5	9	5	10	2	6	7	10
TOKEN PASSING	3	7	5	7	3	4	6	6	8
ROUND ROBIN	8	9	7	8	9	4	8	9	7
MIL-STD-1553B	6	10	7	8	8	5	8	10	7
PRIORITY POLLING	8	6	7	8	5	8	8	8	8
PRIORITY OVERLAY	5	6	6	7	4	8	2	6	3
PRIORITY TIME OUT	7	5	3	7	5	8	5	6	8
FREE ACCESS	5	6	6	7	4	8	4	7	4

Support Synchronous. This assesses the protocols' ability to support a purely synchronous application. Time slots is explicitly designed for this and hence tops the list. A purely synchronous round robin approach is easy to envision and is ranked next, followed by 1553B. The priority protocols are given a medium rating, recognizing that the proper definition of priorities can yield synchronous characteristics. Priority overlay and free access are downgraded slightly because of the bus arbitration process. Token passing is ranked next lowest since portions of the data flow can be skipped for multiple cycles. Collision detection is judged the least able to support synchronous operations.

Support Asynchronous. This assesses the protocols' capability to support asynchronous operations and provide a timely and predictable response to emergency messages. All the priority based approaches are ranked high. Collision detection is rated slightly better than average and 1553B is ranked as average, since a predictable response depends on regular communication with the initiating unit. Token passing and round robin are placed a notch lower since a timely response may depend on which unit is in control. Time slots trail the list for supporting asynchronous operations.

Adaptable to New Technology. This is an estimate of how readily the protocol will accommodate much higher speeds and new technology such as fiber optics. Collision detection and priority overlay are put at the bottom of the list in expectation that the "listen while talk" requirement will be a serious impediment to receiver design and maintenance. Free access is next lowest since the arbitration process appears less predictable/reliable at higher speed. Priority time out is ranked next since the relative efficiency drops off as the speed increases. The other protocols are rated as better than average in accommodating new technology although the synchronization requirements of time slots and the dead bus time of token passing may be impacted by higher speeds.

Similarity to 1553B. This is an estimate of how readily the protocol will be understood and accepted by the user community which is already familiar with MIL-STD-1553B. Round robin essentially just makes use of the dynamic bus control already defined in 1553B. Priority polling just adds the priority field. The free access protocol displays an obvious effort to be "1553B-like". The other protocols do not explicitly infer a command structure, but could certainly be done in a way very similar to 1553B. Only the collision detection is not amenable to a 1553B-like description of the protocol.

Deterministic. This is an assessment of how well the protocol supports a highly predictable flow of data. The time slots protocol heads this list since each and every message in the system can be specified in advance. Several of the priority based protocols conceivably permit the same thing if the message population is small enough. Round robin and 1553B yield better than average predictability impacted mainly the amount of asynchronous messages used. Free access and priority overlay are downrated due to the bus arbitration process and, of course, collision detection is the least deterministic of all the protocols.

Having assigned numeric values to the assorted qualitative judgements made on how the protocols exhibit the desired attributes, an overall assessment of the protocol is generated by summing the number. The resulting ordered ranking is as follows:

1.	1553B	69
2.	Round Robin	69
3.	Priority Polling	66
4.	Time Slots	58
5.	Priority Time Out	54
6.	Free Access	51
7.	Token Passing	49
8.	Priority Overlay	47
9.	Collision Detection	31

It should not be surprising that MIL-STD-1553B comes out at the top of the list. The attributes are aimed at identifying desirable characteristics of a standard. There is, therefore, a built in bias to the existing standard not to mention the explicit attribute of similarity to 1553B.

These numbers are, in effect, a restatement of the conclusions reached before. They endorse, for example, the early decision on ETHERNET, a collision detection protocol. The one slight anomaly displayed is that free access, which is to be given additional attention, ranks behind other protocols which have been dropped from consideration. This is a reminder that free access is being carried forward in the analysis because it is a fairly complete, concrete proposal and is to get the attention of the A2K committee.

8.0 THROUGHPUT ANALYSIS

The next step of the protocol analysis was to assess how the protocol influences the throughput that can be obtained for a given bus rate. This was performed for the three protocols of round robin, priority polling, and free access.

The throughput analysis was based on the following equation:

$$EDT = \frac{N}{A + \frac{20}{R}(N + CS) + G}$$

This equation states that the effective data throughput (EDT) is given by the total number (N) of data words transferred divided by the total time to accomplish the transfer. This total time is expressed as the sum of three terms; the time to acquire the bus (A); the time to transfer all the words of the message including all the data, command, and status words; and finally the gap time (G) embedded in the message sequence.

The time to accomplish the message transfer is formed as the product of the total number of words (N data words plus CS command and status words) and the time to transfer a single word, $\frac{20}{R}$. The R represents the speed of the bus in bits per second and 20 is the assumed number of bit times required for the transfer of one 16 bit word. The use of the 20 means that a 1553B word format is assumed. This is not essential, but since it is applied uniformly across the protocols it should not influence the analysis results.

The throughput analysis began by studying this equation and performing some sample calculations to generate an appreciation of the size of the various terms and thus their relative importance.

One sample calculation used the following parameters:

N = 10 words of data
CS = 2 words of command and status
A = 36×10^{-6} seconds to acquire the bus
G = 4×10^{-6} seconds for gap times
R = 10^7 bits per second on a 10 megabit bus

then:

$$EDT = \frac{10}{36 \times 10^{-6} + \frac{20}{10^7}(10 + 2) + 4 \times 10^{-6}}$$

$$= \frac{10}{40 \times 10^{-6} + 24 \times 10^{-6}}$$

$$= \frac{10^7}{64} = 157 \text{ K words/second}$$

The reader is cautioned that using the units of K words/second absorbs a factor of 10^3. The important observation based on this calculation is that the bus acquisition time of 36 microseconds is the most significant term. This number was actually generated during the early analysis of the free access protocol based on the assumption of nine polling operations requiring four microseconds each (three microseconds for the poll and response plus a 1 microsecond separation)

Further on in the analysis it was estimated that the free access bus arbitration process could be reduced to seven polls of three microseconds each.

Using A=21 microseconds in the above calculation yields:

$$EDT = \frac{10^7}{49} = 204 \text{ K words/second}$$

That is, reducing the acquisition time to 21 microseconds increases the effective throughput by almost thirty per cent. For this reason, the subsequent analysis was careful not to assume the full terminal address field needed to be included in the priority code in order to avoid unnecessarily penalizing free access in comparison with other protocols

To proceed with the throughput comparisons it was necessary to select some of the parameters in the equation. The strategy was to pick as reasonable a value as possible and apply it uniformly across the protocols to avoid introducing any bias.

The first parameter selected was the number of command and status words. Most of the protocols examined generate a need for more commands or status words than the two that typify 1553B. Sometimes an additional command is defined. Free access uses an address word to select the terminal to be active. Often a checksum word is defined. A value of CS=4 was selected as a typical requirement for future high speed protocols.

With this value selected, calculations were done to evaluate the middle term of the denominator for a variety of message sizes and bus rates. The following results (in microseconds) were obtained for the expression $\frac{20}{R}(N + 4)$.

		R =					
		5	10	15	20	30	50
N =	5	36	18	12	9	6	4
	10	56	28	18	14	9	6
	32	144	72	48	36	24	14
	128	528	264	176	132	88	53
	1K	4112	2056	1370	1028	685	411

The principal conclusion drawn from this exercise is that the middle term dominates at low rates and large message sizes. Conversely, the gap time and acquisition time are relatively more important for short messages and high data rates.

It becomes possible now to select a standard gap time. A simple computation for a 100 meter bus yields the conclusion that a minimum of one microsecond is required. An arbitrary choice of two microseconds is then made, confident, based on the above tabulation, that the gap time is not a critical value.

With the number of command and status words fixed at four and the gap time selected to be two microseconds, attentions can be focused on the acquisition time which is the parameter which is most highly protocol dependent.

Again, a number of parameters are involved and again the strategy is to select reasonable values and apply them consistently across the protocols. When bus length is involved, 100 meters will be used. This, in turn, means the "standard gap" of two microseconds may be maintained.

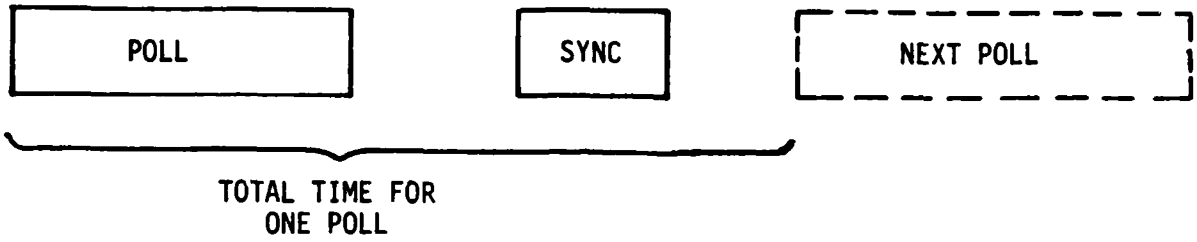
8.1 Bus Acquisition: Free Access

The bus acquisition time for the free access protocol is the number of polls multiplied by the time for each poll, added to a "terminal turnaround time". The terminal turnaround time is the time required for the terminal to discover it has won the contention process, fetch the first bus instruction, decode this and initiate the first command word on the bus. To estimate this, experience with the AN/AYK-15A processors was used. A typical value of 30 microseconds was then scaled by a factor of 10 to account for VHSIC type of technology. The time used for the terminal turnaround time was 3.0 microseconds.

The number of polls required in the free access protocol as modified during this analysis is one (for the "anybody higher" poll) plus the number of bits in the priority code. The priority code is composed of a three bit message priority concatenated with a terminal address. The number of bits required for the terminal address depends, of course, on the number of potential controllers. The relationship of these numbers is as follows:

Number of Controllers	Bits For Terminal Address	Bits in the Priority Code	Total Number of Polls
2	1	4	5
3,4	2	5	6
5-8	3	6	7
9-16	4	7	8
17-31	5	8	9

The time required for each poll in the free access protocol is composed of the time for the poll word, the gap following the poll, the three bit time sync response, and then an intermessage gap preceeding the next poll. Pictorially, this is represented as follows:



The poll and the sync require a total of 23 bit times. The "standard" gap of 2.0 microseconds is used for the response gap preceeding the sync. For the intermessage gap following the sync, a value of 1.0 microseconds is used. This again represents an optimistic "VHSIC-like" estimate. Altogether, this results in the time for polling operation in the free access protocol of:

$$\frac{23}{R} \quad + \quad 3.0 \text{ microseconds}$$

and therefore the total expression for the bus acquisition time in the free access protocol becomes:

$$A = NP * (\frac{23}{R} + 3.0) + 3.0$$

where NP represents the number of polls required which is related to the number of controllers in the system as previously identified.

Carrying out the calculations for a variety of bus rates and number of controllers results in the following tabulation of bus acquisition times for the free access protocol:

NUMBER OF CONTROLLERS	NP	R=					
		5	10	15	20	30	50
2	5	41.0	29.5	25.7	23.8	21.8	20.3
3-4	6	48.6	34.8	30.2	27.9	25.6	23.8
5-8	7	56.2	40.1	34.7	32.0	29.4	27.2
9-16	8	63.8	45.4	39.3	36.2	33.1	30.7
17-31	9	71.4	50.7	43.8	40.3	36.9	34.1

8.2 Bus Acquisition: Priority Polling

The bus acquisition time for the priority polling protocol is the sum of the times for:

- polling (number of polls * time for each)
- algorithm (to identify highest priority)
- handover (dynamic bus control)
- terminal turnaround time

With N controllers in a system, N-1 polls are required, each consisting of a command word, gap time and status word. The polling time is therefore $(N-1)\ (\frac{40}{R} + 2.0)$.

For the algorithm time it is assumed some basic overhead is involved plus process time per controller. The equation used is 1.0 + N*0.5 microseconds, which obviously is again a VHSIC type of assumption. The handover command is a dynamic bus control mode command which means a command gap and status. However, the status word transmission on the bus is overlapped in time with the terminal turnaround time and thus not counted separately. The time attributed to the handover is just 20/R + 2.0. For the terminal turnaround time, the value 3.0 microseconds is used again. The total bus acquisition time for the priority polling protocol is:

$$A = (N-1)\ (\frac{40}{R} + 2.0) + 1.0 + N * 0.5 + \frac{20}{R} + 2.0 + 3.0$$

$$= 4.0 + N\ (\frac{40}{R} + 2.5) - \frac{20}{R} \text{ microseconds}$$

Evaluating this expression for a number of controllers at various bus speed results in the following tabulation of bus acquisition times for the priority polling protocol.

		R=					
		5	10	11	20	30	50
N=	2	21.0	15.0	13.0	12.0	11.0	10.2
	4	42.0	28.0	23.3	21.0	18.6	16.8
	6	63.0	41.0	33.6	30.0	26.3	23.4
	8	84.0	54.0	44.0	39.0	34.0	30.0
	16	168.0	106.0	85.3	75.0	64.7	55.8
	24	252.0	158.0	126.7	111.0	95.3	82.8
	32	336.0	210.0	168.0	147.0	126.0	109.2

8.3 Bus Acquisition: Round Robin

For the bus acquisition time for the round robin protocol it was assumed the terminal offered control via a dynamic bus control command could accept or refuse bus control. The time for the bus acquisition in the round robin protocol is then the sum of:

the number of refusals * time for the offer
the handover command
the terminal turnaround time

The time for the offer of bus control covers a command, gap, status and inter-message gap which is:

$$\frac{40}{R} + 2.0 + 1.0$$

The handover command covers a command, gap and status with the status again overlapped with the terminal turnaround. This then contributes 20/R + 2.0. The terminal turnaround is again taken to be 3.0. Representing the number of refusals as NR, the total bus acquisition time for the round robin protocol then becomes:

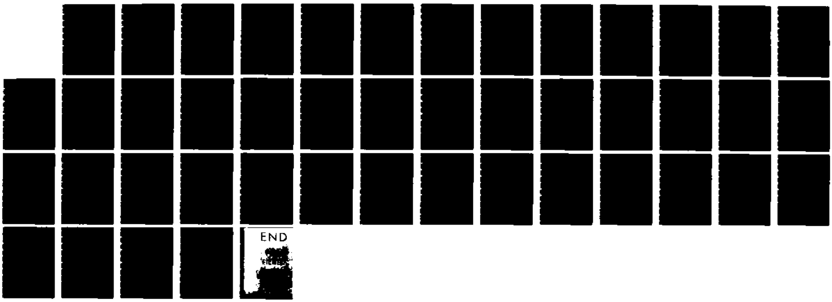
$$A = NR * (\frac{40}{R} + 3.0) + \frac{20}{R} + 2.0 + 3.0$$

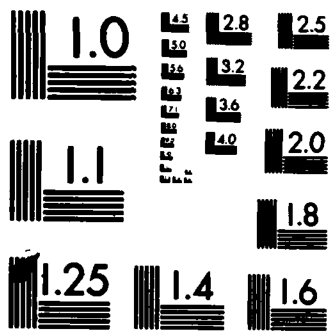
$$= 5.0 + \frac{20}{R} + NR * (\frac{40}{R} + 3.0)$$

Evaluating this for a number of rates and values of NR yields the following tabulation:

		R=					
		5	10	15	20	30	50
NR=	0	9.0	7.0	6.3	6.0	5.7	5.4
	1	20.0	14.0	11.9	11.0	10.0	9.2
	2	31.0	21.0	17.6	16.0	14.4	13.0
	3	42.0	28.0	23.3	21.0	18.7	16.8
	4	53.0	35.0	28.9	26.0	23.0	20.6
	5	64.0	42.0	34.6	31.0	27.4	24.4

AD-A138 226 MULTIBUS AVIONIC ARCHITECTURE DESIGN STUDY (MAADS)(U) 2/2
TRW DEFENSE SYSTEMS GROUP RDDONDO BEACH CA
B A RICH ET AL. OCT 83 AFWAL-TR-83-1141
UNCLASSIFIED F33615-81-C-1520 F/G 17/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

8.4 Protocol Comparisons

As an intermediate step in the throughput analysis the bus acquisition times were compared. Note that the acquisition time basically measures the time interval from when one controller completes its period of bus control until the next controller begins bus operations. To accomplish this comparison it was necessary to relate, for the round robin protocol, the number of refusals to the total number of controllers.

If one were to assume that whenever bus control was offered it was immediately accepted, the obvious result would be an apparently very efficient protocol with extremely low bus acquisition time. This might actually be the case for a brief period of time, say for example, while basic synchronous operations were being performed. But this, it is expected, would persist for only a short period of time, perhaps for one trip around the system. After that, control would be refused until an asynchronous message requirement arose in the system. If, in general, when the dynamic bus control mode code list is started, there exists one asynchronous requirement somewhere in the system then, on the average, half the controllers in the system will be polled before control is accepted.

If a lighter load is projected, then the system "idles" for a while until an asynchronous requirement arises. The time of interest now becomes the interval until the correct controller is offered control. Since control may be anywhere in the system when this occurs, we again on the average, have half the total controllers refuse control before it is offered to the correct one having the asynchronous request. The assumed relationship between the number of controllers and the number of refusals is therefore

$$NR = \frac{N-1}{2}$$

Using this relationship for the round robin protocol and extracting values from the previous tabulations, an aggregate tabulation of bus acquisition time can be made. This is done for the two extremes of the previous tabulations, namely 5 megabits per second and 50 megabits per second.

The result is as follows:

FOR 5 MHz

	FREE ACCESS		PRIORITY POLLING		ROUND ROBIN	
N	NP	A(μS)	N	A(μS)	NR	A (μS)
1						
2	5	41.0	2	21.0		
3	6	48.6			1	20.0
4			4	42.0		
5	7	56.2			2	31.0
6			6	63.0		
7					3	42.0
8			8	84.0		
9	8	63.8			4	53.0
10						
11					5	64.0
12						
13						
14						
15						
16			16	168.0		
17						

FOR 50 MHz

1						
2	5	20.3	2	10.2		
3	6	23.8			1	9.2
4			4	16.8		
5	7	27.2			2	13.0
6			6	23.4		
7					3	16.8
8			8	30.0		
9	8	30.7			4	20.6
10						
11					5	24.4
12						
13						
14						
15						
16			16	55.8		
17	9	34.1				

The graphic representation of these values are shown in Figures 1 and 2.

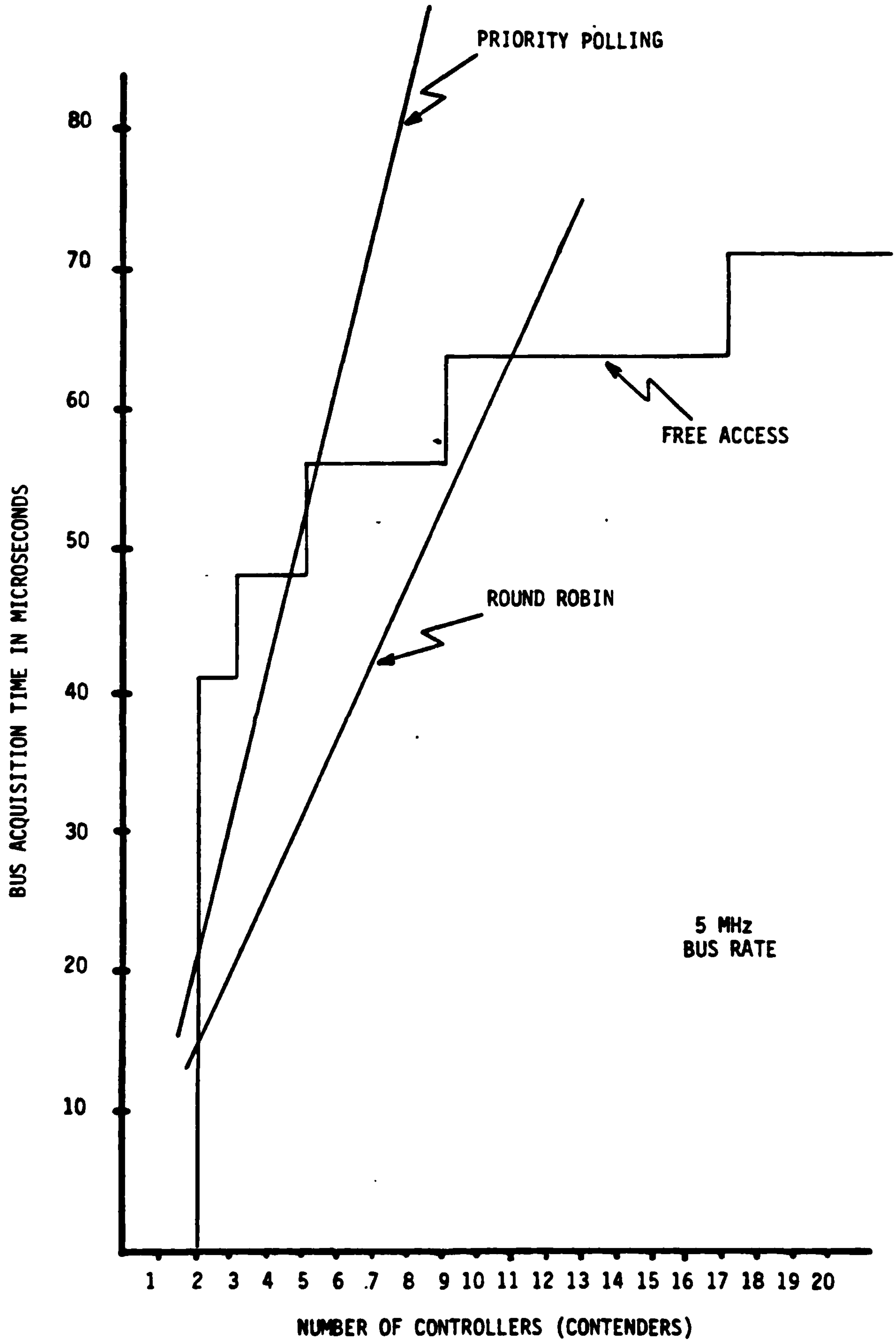


Figure 1. Protocol Comparisons: 5 MHz

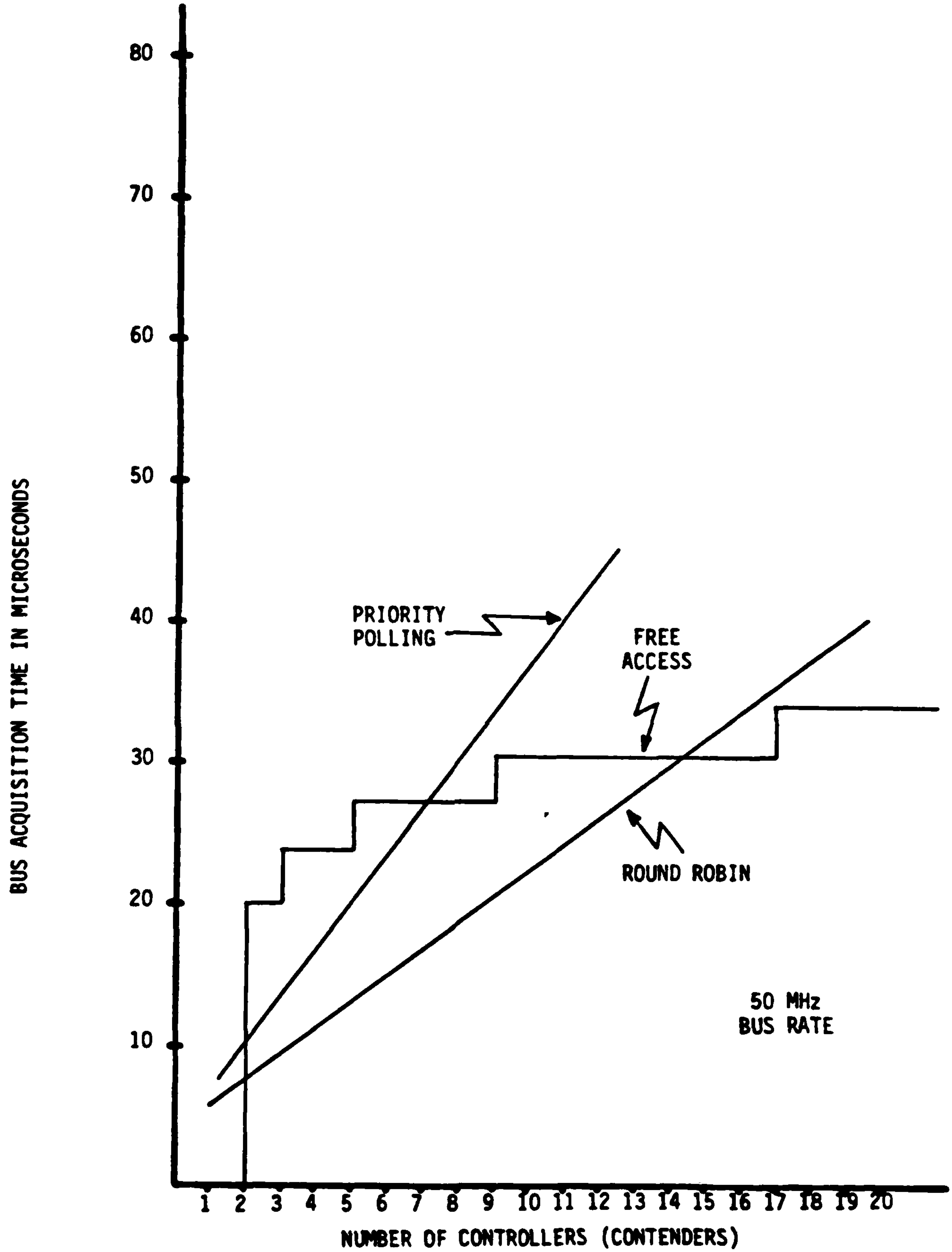


Figure 2. Protocol Comparison: 50 MHz

The most elemental conclusion drawn from these charts is that the general characteristics and relative performance of the protocols persist across the entire range (5 megabits per second to 50 megabits per second) of data rates examined. The conclusions reached should be valid then, independent of the final selection of a rate for the high speed data.

As expected, the free access protocol is the worst of the three for a small number of controllers and the best of the three for a large number of controllers. The crossover point with priority polling is in the range of five to seven terminals. The crossover point with round robin is in the range of eleven to fifteen controllers.

For all rates and any number of controllers the round robin protocol does better than the priority polling protocol. But it is to be recognized the two protocols have a slightly different effect. The priority polling hands over control to the terminal with the current highest priority message. The round robin hands over control to the first terminal that accepts it. With sparsely populated priority class the two approaches yield very nearly the same result. In a more dense message environment the two diverge, with priority polling yielding a steady response to priority messages while the round robin response to a priority message degrades. But at the same time notice that as the message density increases the acquisition time for round robin decreases and therefore the performance advantage the round robin already enjoys grows even larger. The tradeoff between these is clearly a design issue that can only be made with a keen awareness of the message transfer requirements of the system.

To press on with the throughput performance analysis it is necessary to select a particular number of controllers as the typical case to be used as the basis of comparison. It has already been noted in prior discussions that the original free access protocol implication that all terminals would be controllers is not judged to be realistic. (As an aside here, notice that if one endorses that concept as the real problem to be solved, say for up to sixty-four contending controllers, then the preceding charts are making a strong statement that the free access protocol is far and away the best approach.)

The exact number of controllers to be used is a somewhat arbitrary choice. Clearly, two is the minimum. Experience with avionics systems would suggest four as a reasonable number. But to define a standard protocol for future generalized high speed bus applications it seems wiser to work with a slightly larger number. Based on these considerations and the crossover points identified on the preceding charts the choice is made to deal with six controllers as the typical case to be used as the basis for throughput comparisons. This appears to be both enough controllers to represent a future generalized case and also to fall in a region where a fair comparison of the protocols can be made.

Extracting data for the six controller case from previous tabulations, one arrives at the following set of values to be used for the bus acquisition times in throughput calculations.

	R=					
	5	10	15	20	30	50
FREE ACCESS	56.2	40.1	34.7	34.0	29.4	27.2
PRIORITY POLLING	63.0	41.0	33.6	30.0	26.3	23.4
ROUND ROBIN	42.0	28.0	23.3	21.0	18.7	16.8

8.5 Throughput Comparisons

Once the bus acquisition times have been determined, the throughput comparison is a simple application of the equation

$$EDT = \frac{N}{A + \frac{20}{R}(N+4) + 2}$$

This results in the following values of (Kilowords per second) throughput for the indicated message sizes and bus rates

	R=					
MESSAGE SIZE	5	10	15	20	30	50
	FREE ACCESS					
5	53	83	103	111	134	152
10	88	143	181	200	245	287
32	158	281	378	444	578	734
128	218	418	602	762	1072	1561
1024	246	488	728	962	1429	2325
	PRIORITY POLLING					
5	50	82	105	122	147	172
10	83	141	185	217	265	323
32	153	278	383	471	612	804
128	216	417	605	781	1101	1637
1024	246	488	728	966	1435	2346

MESSAGE SIZE	R=					
	5	10	15	20	30	50
	ROUND ROBIN					
5	62	104	134	153	187	223
10	100	172	228	270	332	410
32	170	314	437	542	716	964
128	224	435	636	826	1178	1788
1024	247	491	734	974	1450	2381

Several observations may be made about these results. First it is noted that the round robin protocol is superior in all cases. At the smaller message sizes, and data rates it exhibits about a 20% advantage over the free access protocol. This grows to near 50% for the high rate, short message case and falls back as the message size increases.

In fact, as the message size increases, the differences between all the protocols becomes less and less significant. In general, it can be stated that the message size is a much greater influence on the throughput than anything observed in the protocols. A particular case may be seen in the free access numbers for a 5 word message on a 5 megabit bus. Doubling the message size is more effective than doubling the bus speed. Going to a 32 word message is more effective than a tenfold increase in the bus speed!

Another interesting fact is that the free access and priority polling throughput number crossover just above ten megabits. Free access is more effective below this value and priority polling is better above. This reflects the fact that free access is more sensitive to propagation delay while priority polling depend on word transfer times.

The computed throughput values for the 32 word messages are presented graphically in Figure 3. This again displays the overall superiority of the round robin protocol for throughput.

8.6 Conclusions

At this point in the protocol analysis an intermediate conclusion was reached. It was recognized that while round robin gives the best throughput results it does not give the same priority message response as is available in the priority polling approach. It seems a reasonable conjecture that a system designer might very well choose either approach according to the requirements of a specific system. It is also to be recognized that while these have been described as distinct protocols, they could equally well be viewed as two different control strategies implementable under a single standard protocol. Both, after all, utilize dynamic bus control as the control exchange mechanism.

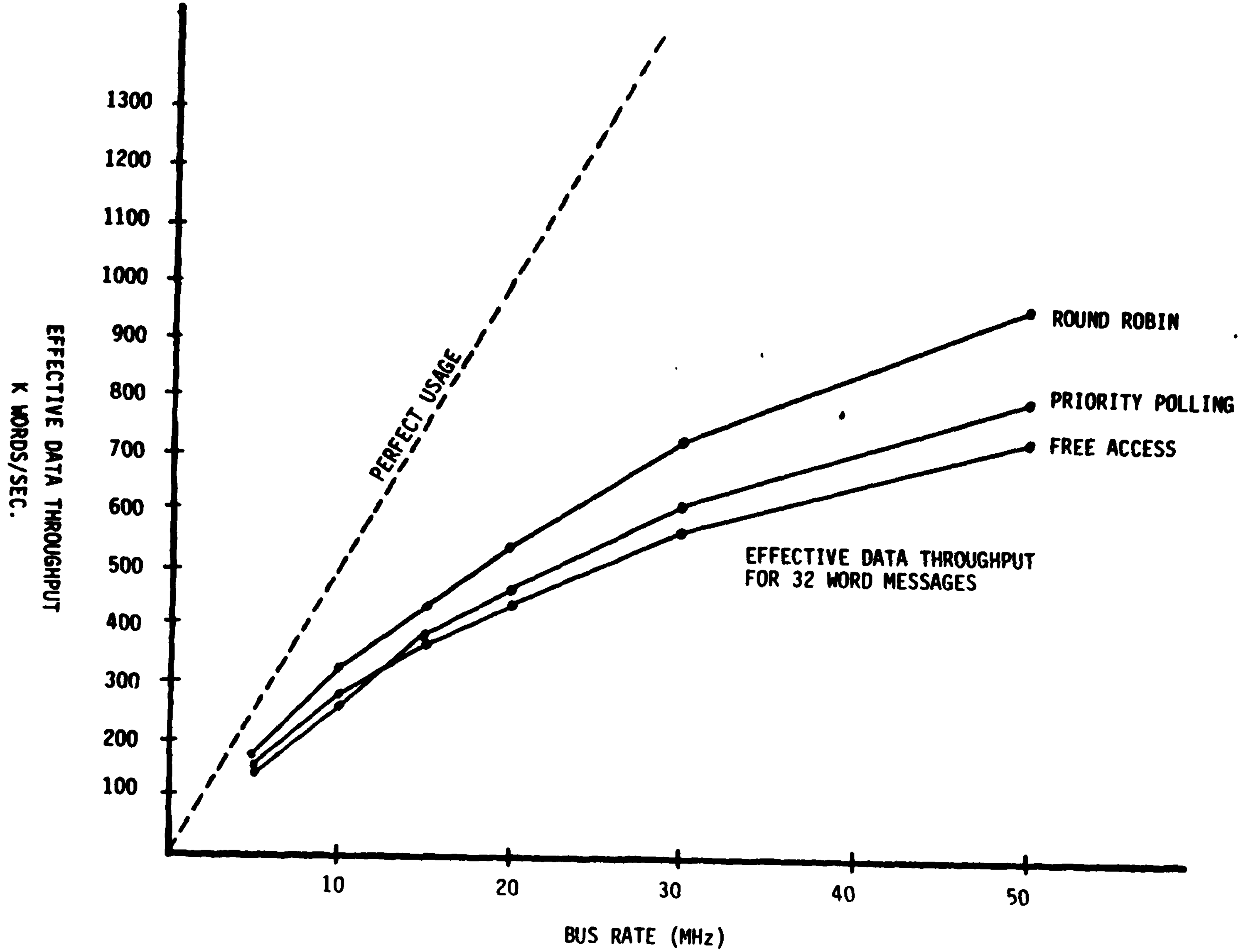


Figure 3. Throughput Comparisons

They just differ in the methodology of deciding when and to whom control is passed. A hybrid of the two approaches (e.g., a once-around round robin for synchronous operation followed by priority polling for asynchronous) is an entirely conceivable possibility.

Similar comments do not apply to the free access protocol. It is a different protocol and a choice must be made. The choice is to drop free access from further consideration. It exhibited a mediocre ranking in the attribute analysis and now has shown itself least effective with respect to throughput.

At this point, the protocol analysis took on a new twist. Having discarded free access and having actively considered a hybrid of priority polling and round robin, the question arises as to whether it is wise to completely ignore the stationary master. Isn't this, in fact, a reasonable candidate for future systems, and actually the best solution for some applications? Particularly in the context of a hybrid approach, it seems useful. The once-around, round robin for synchronous operations mentioned above, actually implies a controller operating as a stationary master for that period of time that it maintains control.

Another motivation for considering the stationary master is that it necessarily has a small bus acquisition time and therefore should have good throughput characteristics. For a stationary master protocol, the bus acquisition time is simply the intermessage gap time. In the context of VHSIC technology, a choice of four microseconds seems reasonable. A lesser value might be justifiable, but that could be viewed as biasing the analysis. The throughput equation for the stationary master then becomes

$$EDT = \frac{N}{4 + \frac{20}{R}(N+4) + 2} = \frac{N}{6 + \frac{20}{R}(N+4)}$$

Performing the calculations for this case results in the following tabulations.

MESSAGE SIZE	R= 5	10	15	20	30	50
5	119	208	278	333	417	521
10	161	294	405	500	652	862
32	213	410	593	762	1067	1569
128	239	474	703	941	1362	2177
1024	249	497	744	990	1481	2454

Comparison with previous tabulation will show that for the shorter message sizes the stationary master nearly doubles the throughput as compared to the other protocols. The data for 32 word messages is presented graphically in Figure 4, overlaid on the previously presented graphs.

At the conclusion of this analysis of the protocols appropriate for a future, standard, high speed bus we thus have;

- stationary master which provides the best throughput;
- round robin, which provides the quickest transfer of control with good fault tolerance characteristics;
- priority polling, which provides the most predictable response to emergency messages.

The high speed bus protocol defined (separately documented as an appendix to the MAADS architecture specification) will accommodate all of these strategies. This will permit the system designer to mix and match the strategies according to the message transfer requirements of the particular system being designed.

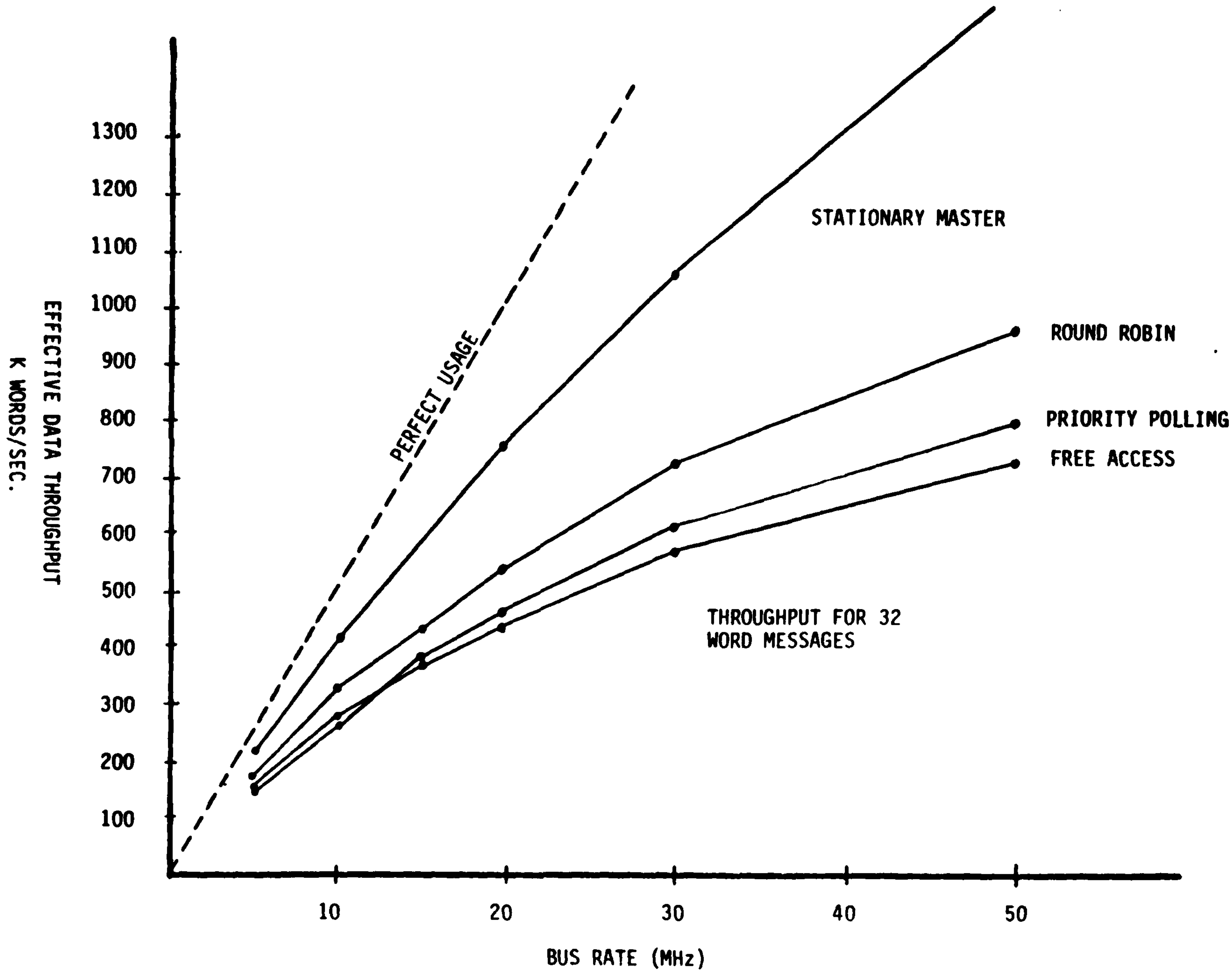


Figure 4. Throughput Comparisons

APPENDIX B

MULTIBUS AVIONICS ARCHITECTURE

DESIGN STUDY (MAADS)

TECHNICAL REPORT

EVALUATION OF HIGH SPEED BUS PROTOCOLS

February 15, 1983

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

In the course of studying advanced avionics architectures, TRW, on the MAADS project, has devoted considerable effort to the analysis and evaluation of protocols suitable for the operation of a high speed multiplex data bus in future avionics systems. A formal review of this effort was conducted with the Air Force in September 1982. This review not only presented a proposed protocol, but also discussed the assorted criteria used in evaluating various protocols.

TRW was requested by the Air Force to pursue the area of evaluation criteria further and develop specific definitions suitable for review by a wide range of personnel. TRW has accomplished this and also identified a systematic methodology for evaluating a protocol against the criteria. Both of these, the criteria and the evaluation method, have been presented to the SAE/AE-9B Committee at its recent meeting. This committee is in the early stages of defining a standard for high speed data buses.

Early in the MAADS effort it was recognized that whenever a standard for a high speed bus was adopted, the AVSAIL would be obliged to use it. The options available to MAADS, therefore, were to: 1) wait for the standard to be defined; 2) attempt to define the standard; or 3) plan on a retrofit to the standard somewhere downstream in the PAVE PILLAR program. The first option is incompatible with the MAADS schedules and objectives. The second option is unrealistic; no one group or project is capable of defining the standard. It takes a mechanism, specifically the AE-9B Committee, which incorporates industry wide participation to accomplish this. The third option is thus to be expected, but is by definition beyond the scope of the MAADS effort.

A pragmatic assessment of this situation yielded the conclusion that the Air Force customer was best served if MAADS were as cooperative as possible with the activities of the AE-9B committee. The MAADS project adopted this position and maintains it in the expectation that this will minimize future retrofit problems.

It is hoped this report will further foster that spirit of cooperation. It must be recognized that the selection of bus criteris is a somewhat arbitrary procedure and the evaluation process, even when approached in a quantitative fashion, is largely a matter of subjective engineering judgement. Bus criteria and the evaluation process gain increased validity, therefore, only when given widespread acceptance.

1.1 Purpose and Scope

This report identifies and documents the criteria TRW used to evaluate high speed bus protocols. The methods of developing and applying the criteria are discussed and a summary of the results of the MAADS evaluation of protocols is presented. This report does not contain a description of all the analysis leading up to the design of the MAADS protocol. That material is to be found in the interim technical report.

2.0 METHODOLOGY

The development of a set of criteria for evaluating high speed bus protocols began during the extensive review of technical literature on bus protocols. Occasionally, in the literature, explicit attention is given to criteria but, more frequently, the criteria are inferred from the objectives or design goals of the protocol. One Air Force commissioned report, the Fault Tolerant Computer Network Study, was particularly useful. This study was performed by IBM and the report explicitly listed evaluation criteria.

This list of criteria was carefully reviewed and adapted to the needs of future avionics system as understood by the MAADS team. To it were added other factors based on engineering judgement and experience. Consideration of various criteria were often as not suggested by the technical literature reviewed on the subject. In all cases a concensus of the engineering personnel involved was achieved.

For the evaluation and weighting process a technique known as fuzzy decision making was employed. This is a computer aided procedure in which a number of criteria must be assessed simultaneously. The criteria are systematically compared in pairs one-on-one. The relative merit of the criteria are then assigned a numeric value indicating the strength of preference for one over the other. When all combinations have been considered, the set of numeric values are input to a small computer program which then cross correlates all the numbers and generates an overall rating of the complete set of criteria. The program also computes and outputs an indication of the internal consistency of the entire set of assessments. This tool was used in the final evaluation of bus protocols with the paired comparison inputs being the concensus of TRW engineering personnel. The MAADS experience is that a group effort arriving at a concensus evaluation was much more effective than a simple compilation of individual evaluations.

This methodology of fuzzy decision making for weighting the high speed data bus evaluation criteria was based on work done in the areas of fuzzy set theory and hierarchial structures. Some of the early work was done by Zadeh. Zadeh defined a fuzzy set as "a class of objects with a continuum of grades of membership". "Such a set is characterized by a membership function which assignes to each object a grade of membership ranging between zero and one."[1] The impetus for defining such things as fuzzy sets is that "many classes of objects encountered in the real physical world do not have precisely defined criteria of membership...and do not constitute classes or sets in the usual mathematical sense of these terms".[1] Examples of fuzzy sets could be the class of expensive homes to the class of underpaid engineers. Zadeh stressed the importance of such sets because "the fact remains that such imprecisely defined 'classes' play an important role in human thinking, particularly in the domains of pattern recognition, communication of information, and abstraction". Zadeh went on to define logical and algebraic operations on fuzzy sets. His paper is one part of the foundation on which the fuzzy decision making methodology rests.

Another part of the foundation is T. L. Saaty's paper, "A Scaling Method for Priorities in Hierarchial Structures". In this paper Saaty investigates a

---

1. Zadeh

method of scaling ratios using the principal eigenvector of a positive pairwise comparison matrix. He also addresses methods for measuring the consistency of the matrix data, and methods for multiple criterion decision making by introducing the notion of a hierarchy. Saaty states that "the hierarchy serves as a useful tool for decomposing a large scale problem, in order to make measurement possible despite the now classical observation that the mind is limited to 7 ± 2 factors for simultaneous comparison". In his work on methods for measuring consistency of the matrix data he points out that "consistency is a necessary but not a sufficient condition for judging how good a set of observational data is. The consistency may be good, but the correspondence of the judgements to reality may be poor." He further states that "a good judge gives good results by any scale including direct estimation. A judge who is not an expert can see in the pairwise comparison process where his judgement is strong and where it is weak. The eigenvalue approach is excellent for bargaining purposes as it permits people to debate the reasons for their estimates, arrive at a consensus, and make compromises here and there."

The bargaining process that Saaty described is the very nature of the approach used by the MAADS team in arriving at a consensus, and it is this same approach and methodology that has been presented to, and basically accepted by, the SEA-AE-9B High Speed Data Bus Committee. The specific algorithm used by MAADS was based on a paper by Yager on "Multiple Objective Decision Making Using Fuzzy Sets". The computer program that was used in weighting and ranking the various bus criteria and bus protocols was derived from Whaley's article on "Fuzzy Decision Making". For those desiring more detailed information on fuzzy sets, hierarchical structures, or decision making algorithms, a bibliography is provided. Zadeh and Saaty's papers are both mathematically rigorous. Yager's paper presents the fuzzy decision making algorithm in a very readable style. Finally, Whaley's paper presents an implementation (written in BASIC) along with a "walk-through" example.

2.1 Bibliography

1. Saaty, T. L. "A Scaling Method for Priorities in Hierarchical Structures," Journal of Mathematical Psychology, 1977, 15, 234-281

2. Whaley, C. P. "Fuzzy Decision Making," Interface Age, November 1979, 87-91

3. Yager, R. R. "Multiple Objective Decision-Making Using Fuzzy Sets," International Journal of Man-Machine Studies, 1977, 9, 375-382

4. Zadeh, L. A., "Fuzzy Sets," Information and Control, 1965, 8, 338-353.

3.0 CRITERIA DEFINITIONS

The high speed bus protocol used by TRW in arriving at its final recommendations are identified and defined in the following sections.

3.1 Fault Tolerance

The capability to endure component errors and/or failures without causing total system failure. An important aspect of fault tolerance is recovery, which includes fault detection, fault containment, fault isolation, and reconfiguration. These are defined as follows:

- Fault detection - ability of a system to determine the occurrence of erroneous operation.
- Fault containment - ability of a system to prohibit errors and/or failures from propagating from the source throughout the system.
- Fault isolation - ability of a system to isolate a failure to the required level so as to be able to reconfigure.
- Reconfiguration - ability of a system to rearrange or reconnect the system elements or functions to provide as near the same system level of operation as before a failure.

3.2 System Integrity

In essence, the degree to which a system is dependable. System integrity will include the following areas:

- Monitorability - ability of the protocol to be viewed passively to allow observation of the dynamics of the protocol in action.
- Testability - addresses how well the protocol supports completeness of testing and facilitates repeatable or predictable results.
- Initialization - support initial configuration of a system on initial powerup.
- Data Link Assurance of Receipt - support assurance of good data through the data link level.

3.3 Throughput/Response

Measure of how well the protocol transfers data from one node's link level to another. Included in this criteria are the following:

- Effective Link Level Data Throughput - throughput of data from data link level to data link level. It is important to distinguish between actual user data throughput as opposed to percentage utilization or loading of the physical transmission medium.

- Data Latency - time delay through transmissing node's data link and physical layers and receiving node's physical and data link layers.

3.4 Message Structure

Addresses issues regarding various capabilities and capacities defined by a protocol relative to the structure of the messages the protocol is designed to handle.

- Addressing Capacity - allows system address expansion directly or indirectly.

- Broadcast Capability - allows messages to be transmitted to all terminals simultaneously.

- Block Transfer - mode to allow transfer of variable length data blocks.

- Content or Labeled Addressing - allow terminals to selectively receive messages based on message labels or message identifiers as opposed to "receive" or "destination" terminal addresses.

3.5 Flexible Network Control Strategy

Addresses how well the protocol leaves the system designer free to address his specific problem (design flexibility).

- Central Control - control from one master, whether stationary or non-stationary.

- Distributed Control - concurrent control from multiple points in the data bus system.

- Support of Synchronous Messages - supports transmission of a series of messages at a known a priori sequence and time or time interval.

- Support of Asynchronous Messages - supports allowing nodes on the data bus to transmit a message whose time of transmission is not known a priori. (Also issue of priority messages requiring immediate access to the bus.)

3.6 Cost/Complexity

Takes into consideration nonrecurring and recurring cost areas, availability of hardware, firmware, and software from commercial sources as opposed to new development in each of these areas.

- Non-Recurring Hardware and Software Costs - cost and complexity of the design and development of the hardware and software necessary to support the protocol.

- Recurring Hardware and Software Costs - cost of the elements in production needed to implement the bus system.

- Support Costs - cost to support the elements of the bus system once they are in the field.

- Weight, Size and Power - measure of the costs needed to meet the physical requirements of the data bus elements.

3.7 Adaptiveness

Addresses how well the protocol lends itself to flexibility.

- Adaptable to New Technology - how easily can the protocol incorporate new technology.

- Compatible with Old Mechanisms - how well can the protocol support elements which are already in existence for current standards (i.e., hardware, software, control strategies).

- Parameterization Capability - how well can the attributes of the protocol be described by parameterizing those elements which can be so structured.

4.0 PROTOCOL EVALUATIONS

Three protocols were identified to be assessed against the defined bus criteria. These consisted of the MAADS protocol, an Ethernet-like protocol, and a token passing protocol.

The MAADS protocol may be described as a fast, but simplified, version of 1553B which makes intelligent use of dynamic bus control and includes a bus inactive timeout to protect against single point failures.

Ethernet is a contention/collision protocol developed for local area networks. By "Ethernet-like" is meant that certain aspects which are designed specifically for commercial applications would be adjusted appropriately for avionics. This was also referred to as an "advanced CSMA/CD" protocol.

A token passing protocol is one in which a control handover takes place frequently so as to form a logical ring structure. It would appear that a token passing design could be implemented within the MAADS protocol, but for the purposes of this evaluation, token passing is considered a separate protocol.

The selection of these three protocols is based on the assumption that they are the ones most likely to be given serious consideration by the AE-9B committee.

The MAADS personnel, in a group effort, then exercised the "fuzzy decision making" process to rank the bus criteria and apply them to the three protocols.

The first step was to perform the paired comparisons of the various criteria. This was done in a concensus fashion and resulted in the numerical values shown in Table 1. The negative values in the table are simply a notational way of indicating a preference for the second of the two criteria being compared. The qualitative interpretation of the numerical values is as follows:

Table 1. Paired Comparison Matrix
Top Level Criteria

	SYSTEM INTEGRITY	THROUGHPUT/RESPONSE	MESSAGE STRUCTURE	FLEXIBLE NETWORK CONTROL STRATEGY	COST/COMPLEXITY	ADAPTIVENESS
FAULT TOLERANCE	3	8	7	6	9	5
SYSTEM INTEGRITY		5	4	-3	3	4
THROUGHPUT/RESPONSE			3	-4	3	-3
MESSAGE STRUCTURE				-5	-2	-3
FLEXIBLE NETWORK CONTROL STRATEGY					-2	1
COST/COMPLEXITY						2

1. Criteria are of equal importance.
3. One criteria is weakly preferred over the other.
5. One criteria is strongly preferred over the other.
7. One criteria is demonstratably preferred over the other.
9. One criteria is absolutely preferred over the other.

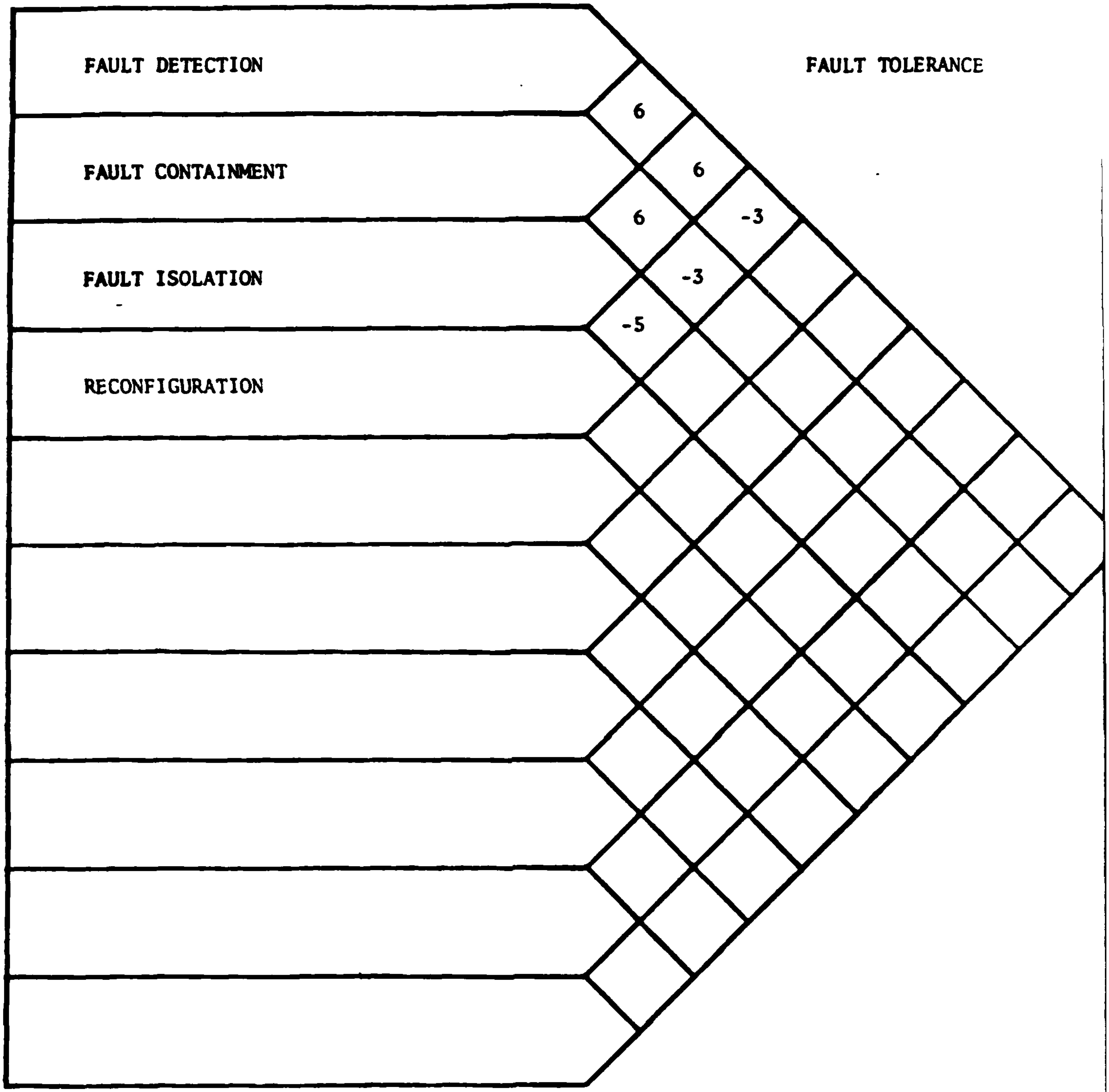
The table, therefore, says that in the opinion of this group of engineering personnel, the criteria of flexible Network Control Strategy and adaptiveness are of equal importance while system integrity is very strongly preferred over throughput. The table also shows that this group considers fault tolerance to be more important than all other criteria.

The next step was to perform the paired comparisons at the subcriteria level. These were then input to the fuzzy decision making (FDM) program and run with a dummy "Nothing" evaluation to get the relative weighting of the subcriteria. The weights are listed under the title "eigenvector" on the printouts. The paired comparison matrices and output weightings are shown in Table 2. This shows, for example, that the second subcriteria under fault tolerance, that of fault containment, is assigned a weight of 0.1246.

The evaluation of the three candidate protocols was then accomplished by rating each of the protocols against each subcriteria on a scale of 0.0 (does not satisfy criteria at all) to 1.0 (satisfies criteria perfectly). These ratings as assigned by the MAADS personnel are shown in Table 3. The parenthetical numbers are the relative weights as developed in Table 2. The resultant rating against the top level criteria are shown as the "composite rating".

The final result of inputing these ratings of the three protocols against the seven top level criteria is shown in Table 4.

Table 2. Subcriteria



P A I R E D   C O M P A R I S O N   M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

FDM PROGRAM OUTPUT
****

EIGENVALUE = 5.35483

EIGENVECTOR ...

.388709 .124558 .0402415 .446491

ALPHA-VECTOR ...

1.55484 .498234 .160966 1.78596

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .47519

WEIGHTED FUZZY SETS ...

.340367

.707973

.894426

.289983

DECISION VALUES ...

Nothing - .289983

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Fault Tolerance

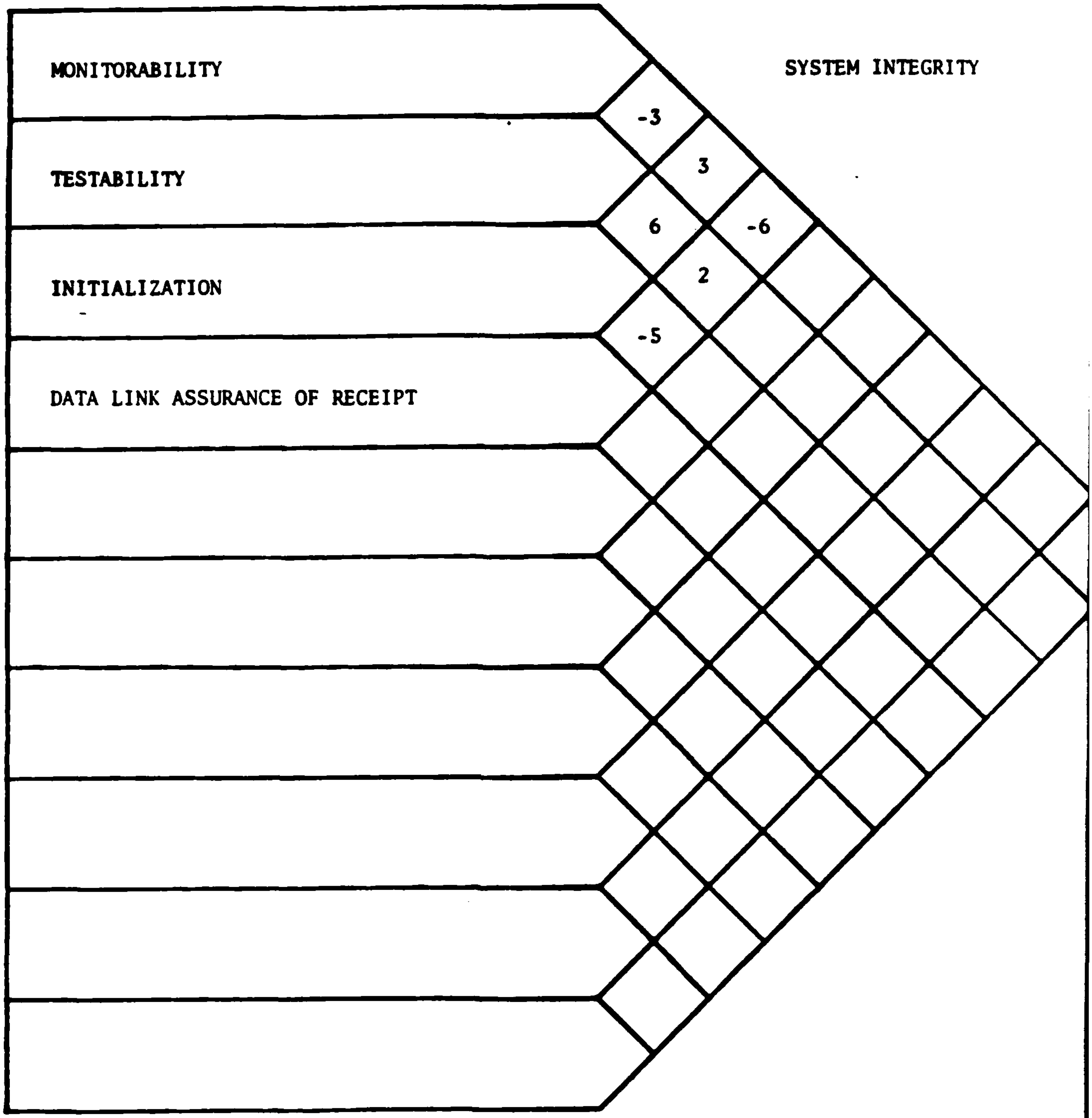
Fault Detection

Fault Containment

Fault Isolation

Reconfiguration

Table 2. Subcriteria (Con't)



P A I R E D   C O M P A R I S O N   M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

Table 2. Subcriteria (Con't)

FDM PROGRAM OUTPUT
****

EIGENVALUE = 4.26092

EIGENVECTOR ...

.118601 .446941 .0580935 .376365

ALPHA-VECTOR ...

.474403 1.78777 .232374 1.50546

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .208535

WEIGHTED FUZZY SETS ...

.719765

.28962

.851233

.352218

DECISION VALUES ...

Nothing - .28962

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

System Integrity

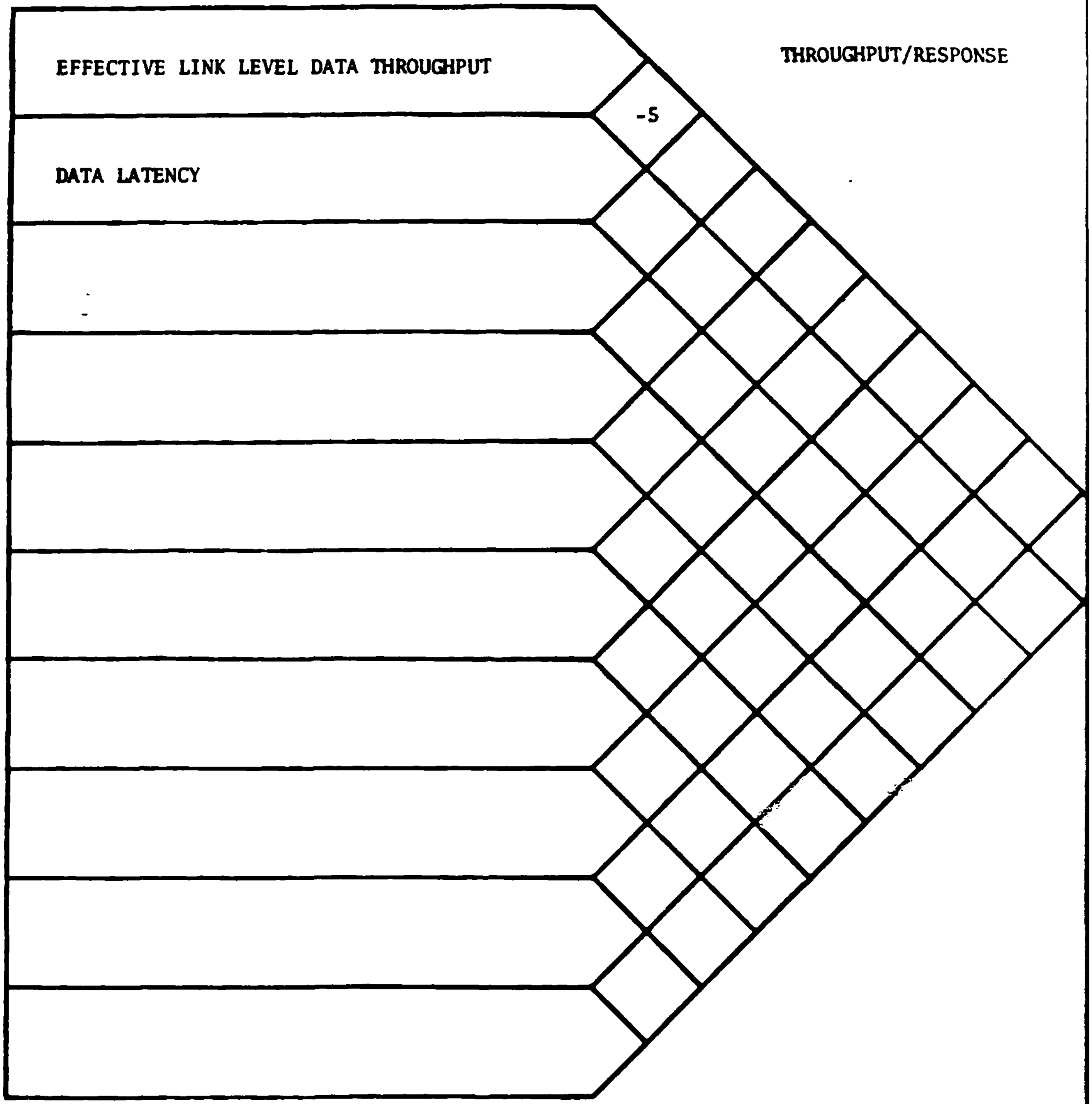
Monitorability

Testability

Initialization

Data Link Assurance of Receipt

Table 2. Subcriteria (Con't)



P A I R E D   C O M P A R I S O N   M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

Table 2. Subcriteria (Con't)

FDM PROGRAM OUTPUT
****

EIGENVALUE = 2

EIGENVECTOR ...

.166667   .833333

ALPHA-VECTOR ...

.333333   1.66667

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = Ø

WEIGHTED FUZZY SETS ...

.7937Ø1

.31498

DECISION VALUES ...

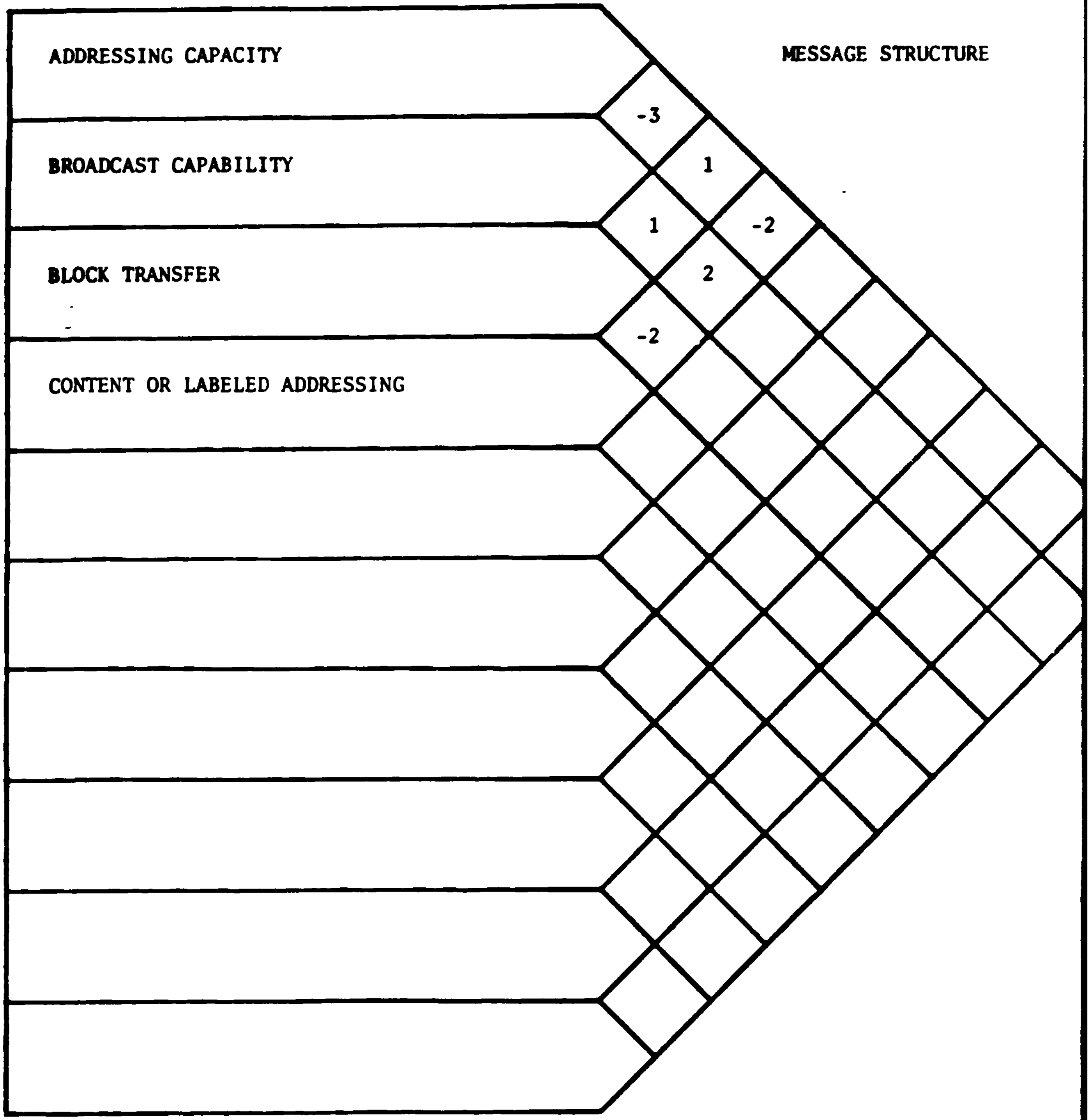
Nothing - .31498

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Throughput/Response

Effective Link Level Data Throughput

Data Latency



P A I R E D   C O M P A R I S O N   M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

Table 2. Subcriteria (Con't)

FDM PROGRAM OUTPUT
****

EIGENVALUE = 4.2Ø8

EIGENVECTOR ...

.145823 .37245 .2Ø4836 .276892

ALPHA-VECTOR ...

.58329 1.4898 .819342 1.1Ø757

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .186189

WEIGHTED FUZZY SETS ...

.66744

.356Ø62

.5667

.464Ø76

DECISION VALUES ...

Nothing - .356Ø62

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Message Structure

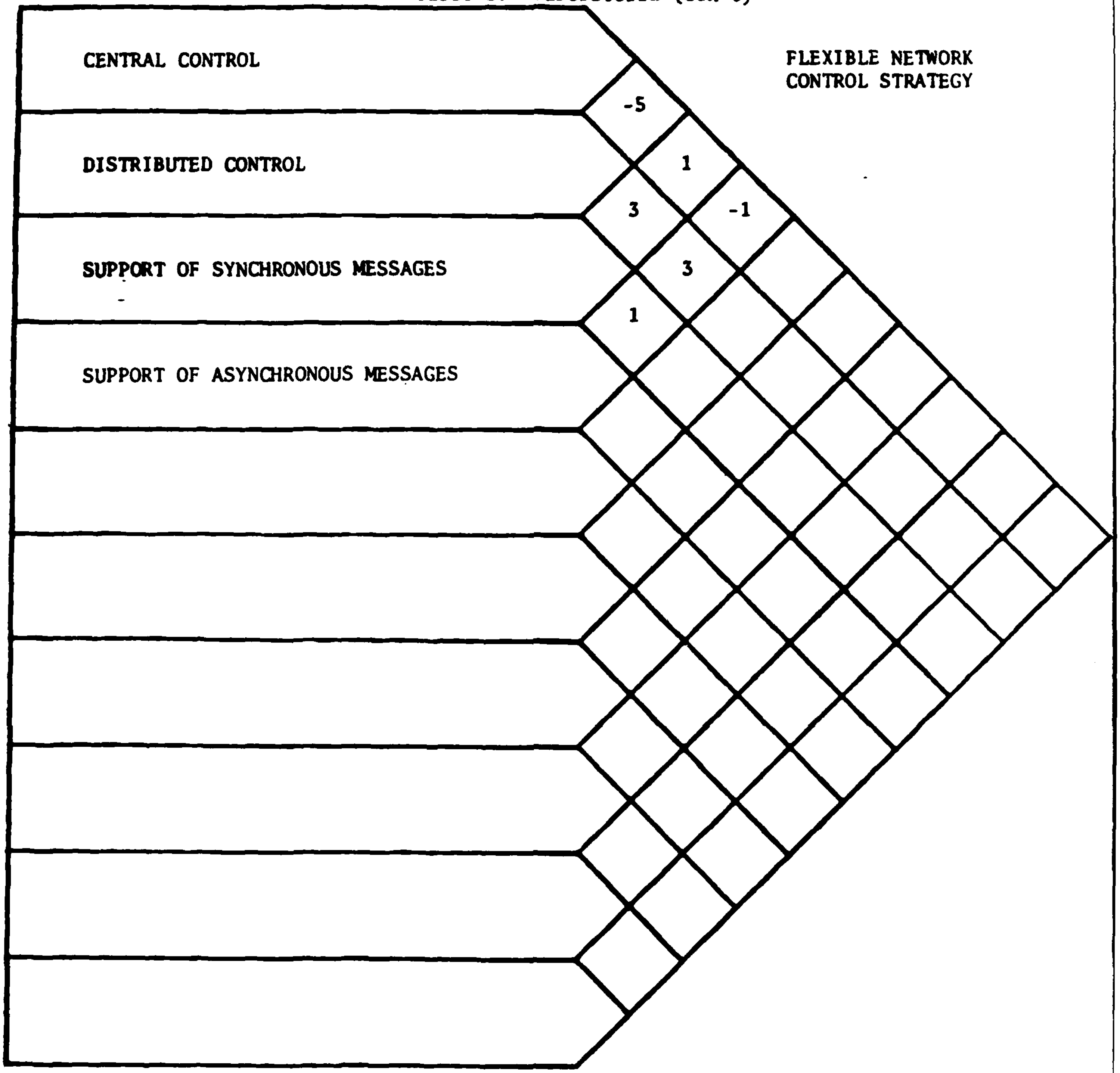
Addressing Capacity

Broadcast Capability

Block Transfer

Content or Labeled Addressing

Table 2. Subcriteria (Con't)



P A I R E D   C O M P A R I S O N   M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

FDM PROGRAM OUTPUT
****

EIGENVALUE = 4.ø2424

EIGENVECTOR ...

.14øø6 .543675 .158133 .158133

ALPHA-VECTOR ...

.56ø241 2.1747 .63253 .63253

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .ø635648

WEIGHTED FUZZY SETS ...

.678189

.221488

.645ø44

.645ø44

DECISION VALUES ...

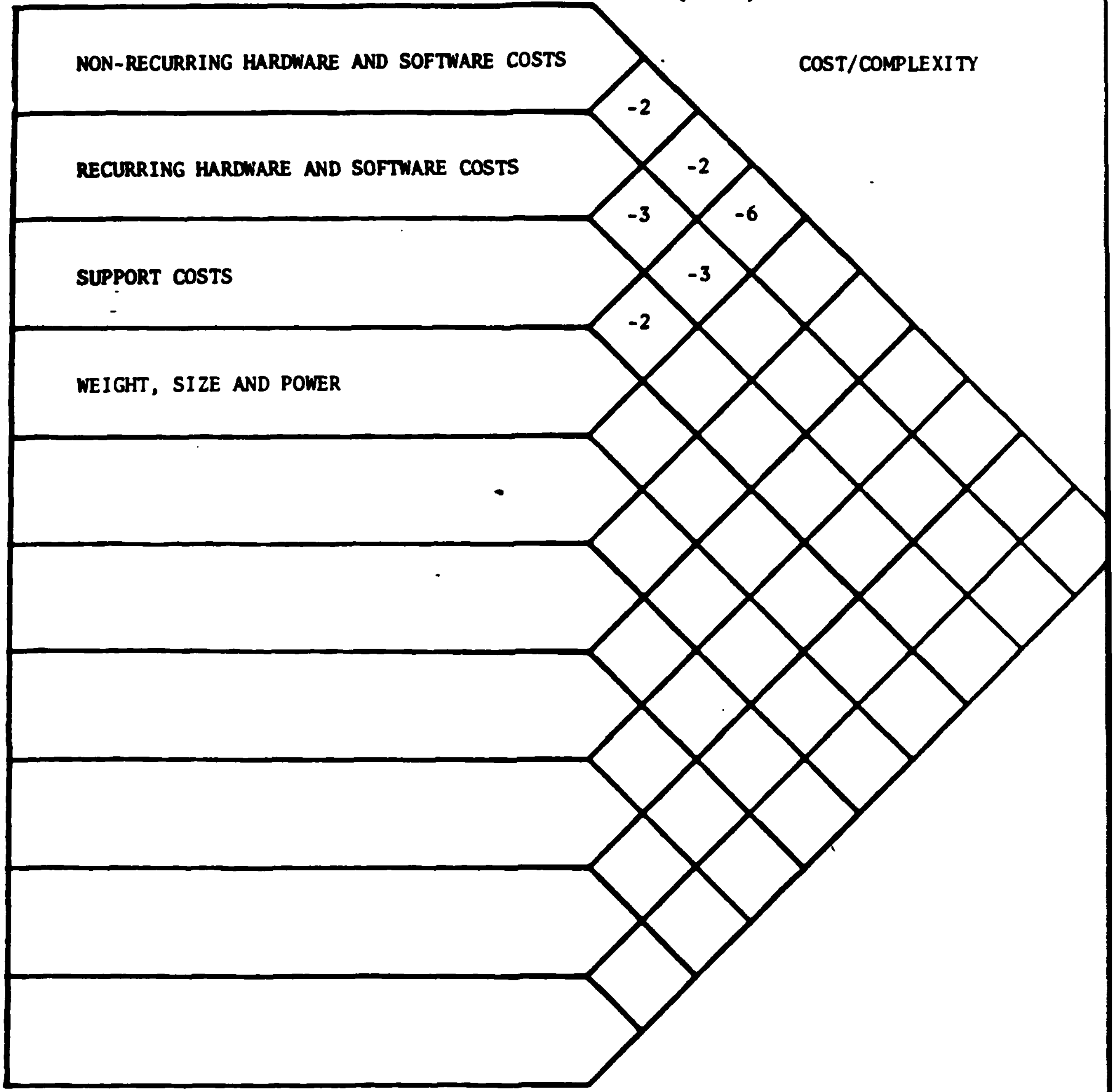
Nothing - .221488

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Flexible Network Control Strategy

- Central Control
- Distributed Control
- Support of Synchronous Messages
- Support of Asynchronous Messages

Table 2. Subcriteria (Con't)



P A I R E D C O M P A R I S O N M A T R I X

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

Table 2. Subcriteria (Con't)

FDM PROGRAM OUTPUT
****

EIGENVALUE = 4.11665

EIGENVECTOR ...

.0930874 .141213 .274618 .491082

ALPHA-VECTOR ...

.37235 .564852 1.09847 1.96433

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .139435

WEIGHTED FUZZY SETS ...

.772523

.676025

.467011

.256259

DECISION VALUES ...

Nothing - .256259

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Cost/Complexity

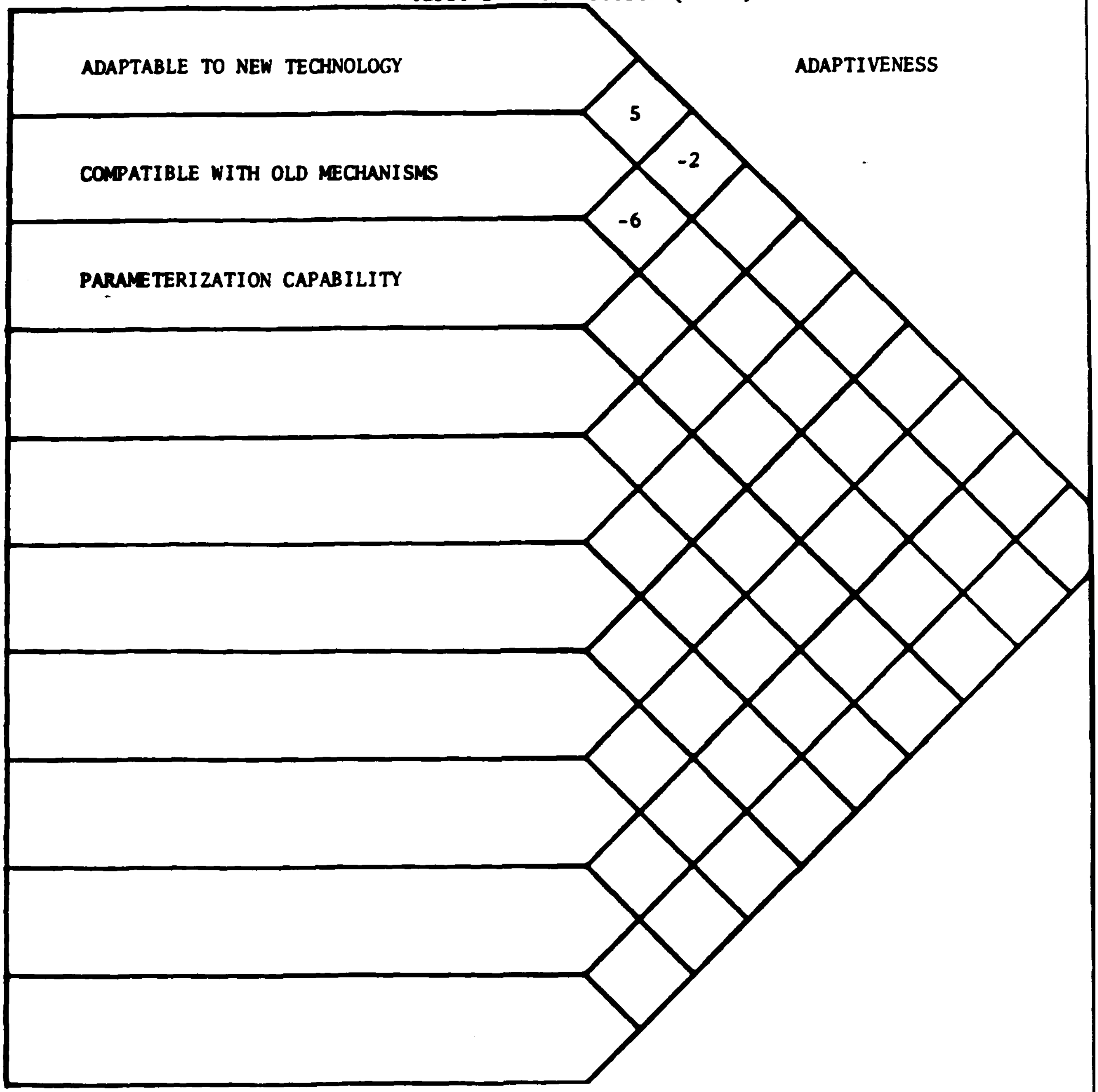
Non-Recurring Hardware and Software Costs

Recurring Hardware and Software Costs

Support Costs

Weight, Size and Power

Table 2. Subcriteria (Con't)



PAIRED COMPARISON MATRIX

Rating scale for Paired-Comparison judgements of the criteria:

Degree of Importance------- Definition

1. Criteria are of equal importance
3. One criterion is weakly preferred over the other
5. One criterion is strongly preferred over the other
7. One criterion is demonstratably preferred over the other
9. One criterion is absolutely preferred over the other

2,4,6,8. Intermediate values; believe two adjacent judgements

Table 2. Subcriteria (Con't)

```
FDM PROGRAM OUTPUT
**************************************************

EIGENVALUE =  3.ø1533

EIGENVECTOR ...

 .341593   .ø811ø97   .577298

ALPHA-VECTOR ...

 1.ø2478   .243329   1.73189

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX =  .ø618987

WEIGHTED FUZZY SETS ...

 .491486

 .844794

 .3ø1ø57

DECISION VALUES ...

Nothing -  .3ø1ø57

Nothing IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.
```

Adaptiveness

Adaptable to New Technology

Compatible with Old Mechanisms

Parameterization Capability

Table 3. Protocol Ratings Against Top Level Criteria

		Advanced CSMA/CD	MAADS	Token Passing
FAULT TOLERANCE				
Fault Detection	(.3887)	0.3	0.8	0.7
Fault Containment	(.1246)	0.6	0.8	0.4
Fault Isolation	(.0402)	0.1	0.8	0.5
Reconfiguration	(.4465)	0.8	0.6	0.3
	Composite Rating	.5526	.7107	.4760
SYSTEM INTEGRITY				
Monitorability	(.1186)	0.1	0.8	0.5
Testability	(.4469)	0.1	0.8	0.7
Initialization	(.0581)	0.8	0.8	0.5
Data Link Assurance of Receipt	(.3764)	0.2	0.8	0.5
	Composite Rating	.1783	0.8	.5894
THROUGHPUT/RESPONSE				
Effective Link Level Data Throughput	(.1667)	0.3	0.6	0.7
Data Latency	(.8333)	0.7	0.6	0.5
	Composite Rating	.6333	0.6	.5333
MESSAGE STRUCTURE				
Addressing Capacity	(.1458)	0.8	0.5	0.8
Broadcast Capability	(.3725)	0.8	0.8	0.7
Block Transfer	(.2048	0.8	0.8	0.8
Content or Labeled Addressing	(.2769)	0.8	0.8	0.8
	Composite Rating	0.8	.7563	.7628

Table 3. Protocol Ratings Against Top Level Criteria (Con't)

FLEXIBLE NETWORK CONTROL STRATEGY

	Advanced CSMA/CD	MAADS	Token Passing
Central Control (.1401)	0.1	0.9	0.6
Distributed Control (.5437)	0.9	0.6	0.7
Support of Synchronous Messages (.1581)	0.1	0.8	0.8
Support of Asynchronous Messages (.1581)	0.8	0.6	0.5
Composite Rating	.6456	.6737	.6702

COST/COMPLEXITY

	Advanced CSMA/CD	MAADS	Token Passing
Non-Recurring H/W and S/W Costs (.0931)	0.8	0.4	0.6
Recurring H/W and S/W Costs (.1412)	0.7	0.4	0.5
Support Costs (.2746)	0.5	0.7	0.6
Weight, Size and Power (.4911)	0.7	0.5	0.6
Composite Rating	.6544	.5315	.5859

ADAPTIVENESS

	Advanced CSMA/CD	MAADS	Token Passing
Adaptable to New Technology (.3416)	0.2	0.7	0.8
Compatible with Old Mechanisms (.0811)	0.1	0.9	0.4
Parameterization Capability (.5773)	0.8	0.6	0.4
Composite Rating	.5383	.6585	.5366

Table 4. Final Evaluation Output

FDM PROGRAM OUTPUT
****

EIGENVALUE = 8.49Ø5

EIGENVECTOR ...

.4Ø69Ø8 .168367 .Ø69Ø717 .Ø3Ø3Ø95 .15Ø917 .Ø917287 .Ø826976

ALPHA-VECTOR ...

2.84836 1.17857 .4835Ø2 .212167 1.Ø5642 .6421Ø1 .578884

CONSISTENCY OF THE PAIRED-COMPARISON MATRIX = .352432

WEIGHTED FUZZY SETS ...

.184627	.378Ø5	.12Ø7Ø1
.131Ø48	.768749	.5363Ø5
.8Ø1822	.781152	.737888
.95376	.94246	.944173
.629857	.658854	.655238
.761645	.666416	.7Ø9446
.6987Ø6	.785172	.697428

DECISION VALUES ...

Advanced CSMA/CD - .131Ø48
MAADS - .378Ø5
Token Passing - .12Ø7Ø1

MAADS IS THE BEST CHOICE ACCORDING TO THE DATA YOU HAVE ENTERED.

Top Level

Fault Tolerance
System Integrity
Throughput/Response
Message Structure
Flexible Network Control Strategy
Cost/Complexity
Adaptiveness

5.0 ELECTRICAL CHARACTERISTICS EVALUATION

Several key areas of concern with the electrical characteristics of the High Speed Bus have been further investigated. These include:

- the feasibility of attaining the operating speeds with present-day integrated circuit technology required to meet the 20 Mbps transmission rate

- the proper selection of a transmission medium

- the mechanism used to establish and maintain clock synchronization

As a result of this additional study effort, the MAADS High Speed Bus Specification was updated to incorporate the study outputs (see Appendix A). In addition, an approach is presented herein to the implementation of an HSB bus controller/remote terminal.

Operating Speed

The MAADS HSB Specification requires a 20 Million bits per second transmission rate, which is a twenty-fold increase over the MIL-STD-1553B rate (1 Mbps). The bus interface units (BIUs) for 1553B are typically implemented with CMOS or low-power schottky (TTL-LS) integrated circuit technology. Samplin rates for this bus range from 8-16 MHz to achieve the required bit error rates. An acceptable sampling rate for the MAADS HSB would be on the order of 200 MHz. This is clearly out of the reach of CMOS or TTL technology, but can be obtained using 100K series ECL technology. And since only the front-end bit synchronization and parallel/serial conversion processes need to attain this excessive speed, a mix of Advanced Low-power (ALS) Schottky TTL and ECL could be employed. The implementation of a BIU out of standard SSI/MSI logic components should include an optimal mix of these two technologies to minimize the power consumption of the device. Unfortunately, power consumption would still remain unacceptably high for anything but a laboratory breadboard unit (20-40 Watts).

Another very viable alternative exists, however. Custom and semicustom integrated circuits have emerged in the marketplace as a much more affordable technology than they were in the past. Many companies offer products in this area ranging from fully customized LSI with high circuit densities, low power consumption, and very high speed operation, to semicustom gate arrays offering reduced prototyping costs and turn-around times at the expense of higher power consumption and circuit complexity. This approach is recommended for implementing the BIUs after design validation using an SSI/MSI breadboard. As a benchmark for comparison, TRW has investigated the feasibility of a 40 Mbps fiber optic bus and sized the preliminary design of the required BIU circuitry (Figure 1). Using the gate array technology, an intelligent interface could be implemented with three major components and five support circuits (Figure 2). The total power dissipation would be less than ten Watts (Table 5).

The TBDs in the MAADS HSB Specification in the section on waveform characteristics are highly dependent on the design of the transmitter/receiver circuit design. Due to the fact that a preliminary design was not feasible within the time constraints of this evaluation, these parameters were left unspecified.

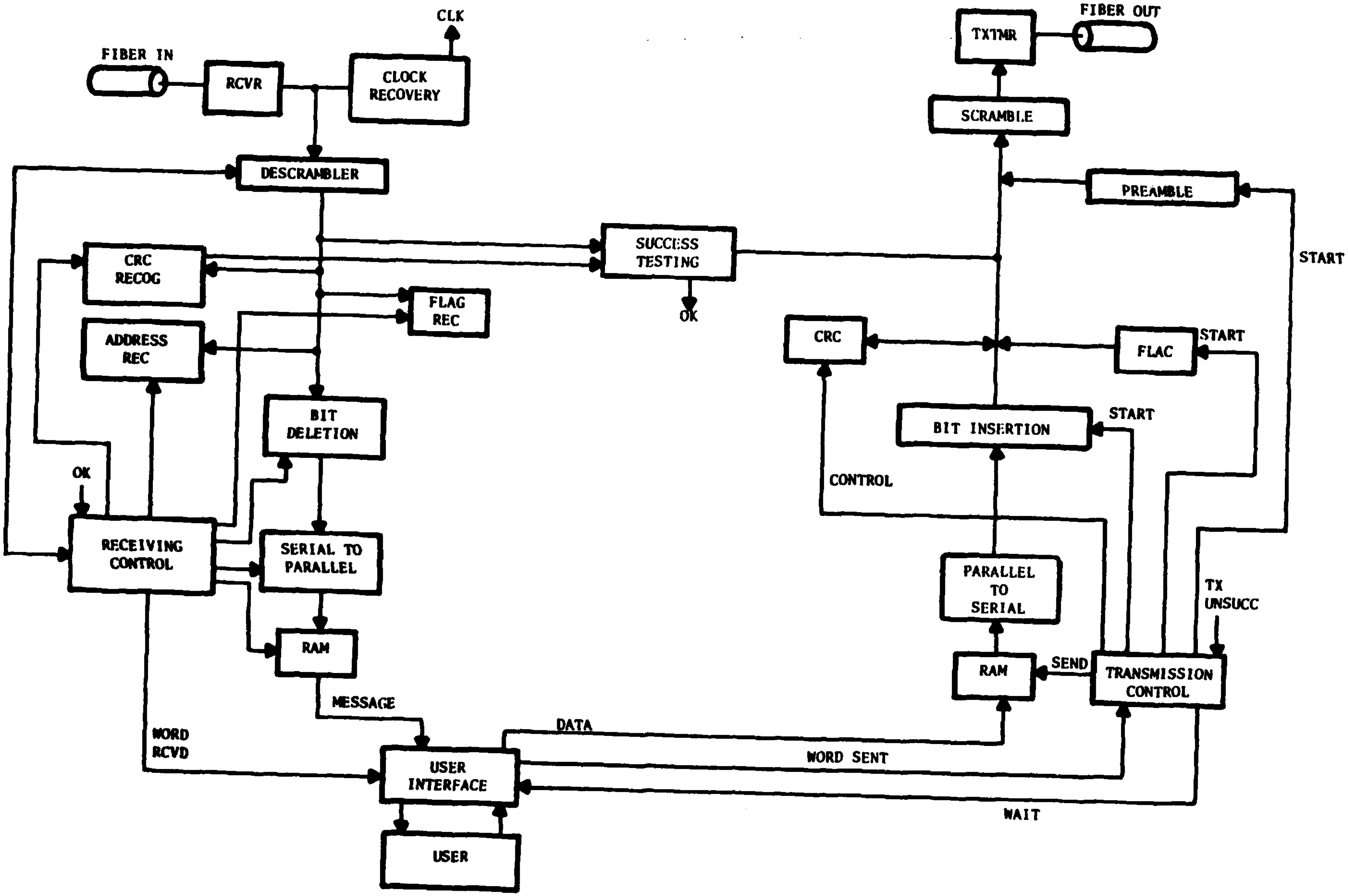


Figure 1. Bus Interface Unit

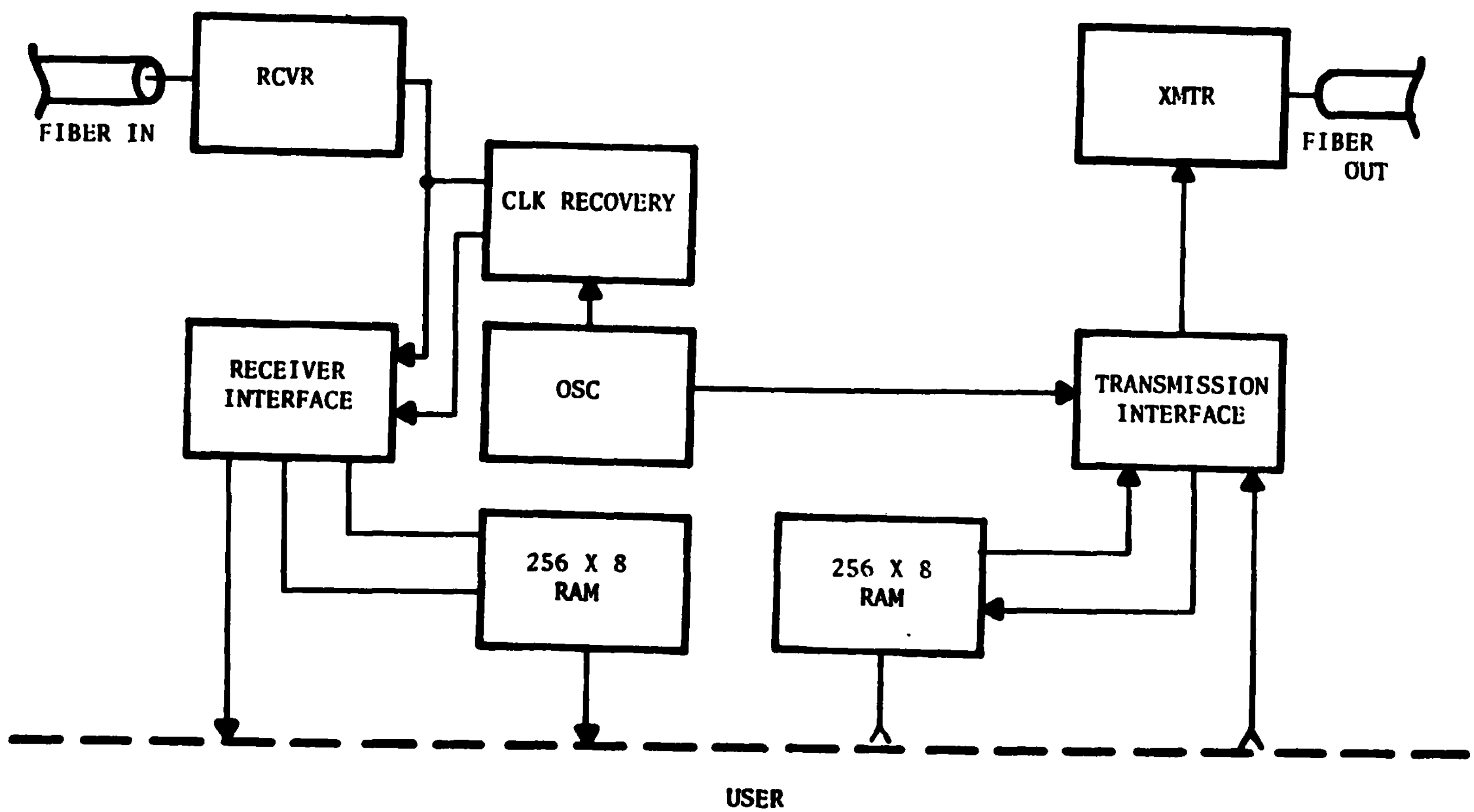


Figure 2. Bus Interface Unit - Physical Partitioning

Table 5. Custom Chip Sizing

CHIP	TECHNOLOGY	TRANSISTOR COUNT	POWER (W)
Transmission Interface	1 3D	12,700	2.6
Receiver Interface	1 3D	14,100	2.9
Clock Recovery	OAT-2	1,500	1.2

Transmission Medium

With the 20 Mbps data being resampled at 200 MHz, it is very important to maintain the integrity of the data edges (i.e., rise and fall times). Because of this, twinaxial transmission lines are not suitable. Instead, it is recommended that a quality grade of coaxial cable be used (such as low-loss Goretex, RG316, or RG400). The penalty for using coax, however, is added weight (especially if double shielded coax is specified).

Clock Synchronization

Clock synchronization for the MAADS HSB could be provided using two different bit synchronization methods. The first is via Manchester encoding of the serial data stream. In this method a transition is forced in the middle of the bit period with the direction of the transition indicating the polarity of the bit. This provides an excellent mechanism to allow the receiver to sync-up on the transmitted signal. Normally, Manchester encoding is used with bipolar signals on differential transmission lines to eliminate the effects of common mode noise. However, when transitioning the HSB from a coaxial cable technology to fiber optic cable technology, a unipolar signal structure must be utilized. Manchester encoding is very straight forward and easy to implement, which means the power required for the encoding/decoding circuitry is very low.

The second method is to append a preamble to the data stream which contains a known pattern (usually alternating ones and zeroes) on which the receiver circuitry can synchronize. This method will lose synchronization if the transmitted data stream is long and no bit stuffing or encoding is employed to assure that there are a sufficient number of transitions at any point in the transmission. The circuitry needed to implement a preamble bit synchronizer requires much more real-estate than the Manchester encoding technique and would consume much more power. Also, when the data stream is extremely short, the preamble wastes a lot of bandwidth. This would be the case for all of the data transfer commands to a transmitting RT, as well as all of the mode commands. For these reasons, Manchester encoding is the recommended approach for HSB clock synchronization.

☆ U.S. GOVERNMENT PRINTING OFFICE 1984—759-062/652

END

FILMED

3-84

DTIC